SAS Publishing

SAS® Inventory Replenishment Planning 9.1.2

User's Guide

The Power to Know®

The correct bibliographic citation for this manual is as follows: SAS Institute Inc. 2004. *SAS® Inventory Replenishment Planning 9.1.2 User's Guide*. Cary, NC: SAS Institute Inc.

SAS® Inventory Replenishment Planning 9.1.2 User's Guide

ISBN 1-59047-528-3

SAS Institute Inc., SAS Campus Drive, Cary, North Carolina 27513.

1st printing, March 2004

Contents

Acknowledgments

Credits

Documentation

Writing	Tugrul Sanli, Jinxin Yi, Michelle Opp, Rob Pratt, Charles B. Kelly
Editing	Virginia Clark
Documentation Support	Tim Arnold, Michelle Opp
Review	Feng Chen, Radhika V. Kulkarni, Michelle Opp, Bengt Pederson, Rob Pratt

Software

SAS Inventory Replenishment Planning was implemented by the Operations Research and Development Department. Substantial support was given to the project by other members of the Analytical Solutions Division.

IRP Procedure	Tugrul Sanli
MIRP Procedure	Jinxin Yi

Support Groups

Software Testing	Bengt Pederson, Rob Pratt, Feng Chen
Technical Support	Tonya Chapman

Acknowledgments

Many people have been instrumental in the development of SAS Inventory Replenishment Planning. We would like to thank Duke University Professor Paul Zipkin for his invaluable comments.

The final responsibility for the SAS System lies with SAS Institute alone. We hope that you will always let us know your opinions about the SAS System and its documentation. It is through your participation that SAS software is continuously improved.

What's New in SAS Inventory Replenishment Planning 9, 9.1, and 9.1.2

Contents

What's New in SAS Inventory Replenishment Planning 9, 9.1, and 9.1.2

Overview

SAS Inventory Replenishment Planning enables you to calculate periodic-review inventory replenishment policies using information about demand, lead time, costs, and desired service measures. For Version 9, IRP existed as a collection of macros; in Release 9.1, these have been extended to the IRP procedure, which can be used to calculate inventory replenishment policies for both single-location and two-echelon distribution inventory systems. SAS Inventory Replenishment Planning 9.1.2 includes the experimental MIRP procedure for calculating inventory control parameters in multi-echelon supply chains. The MIRP procedure is designed to help users better understand how network topologies, cost structures, and service level requirements, among other factors, impact the inventory investment and allocation across a supply chain.

PROC IRP

PROC IRP can calculate two types of replenishment policies, (s, S) policies and (s, nQ) policies. These policies are determined through a number of algorithms that are controlled by user-specified options. PROC IRP can accommodate both single-location and two-echelon distribution inventory systems.

The input data set to PROC IRP specifies information about lead time, demand, and costs, as well as options to control the policy. The output data set produced by PROC IRP gives the policy parameters for each item. In addition, estimates of measures such as fill rate, ready rate, and average inventory, among others, are included in the output data set.

PROC MIRP (Experimental)

PROC MIRP can calculate inventory control parameters for all stock-keeping locations in general supply chain networks, which may consist of any combination of serial, assembly, and distribution subnetworks. These parameters ensure that service requirements are satisfied at minimum inventory cost for every supply chain. PROC MIRP currently supports the base-stock replenishment policy and the ready rate as the service level requirement.

The input data sets to PROC MIRP contain information about lead times, holding costs, service levels, and demand of stock-keeping location in the networks, as well as information about the structure of the networks. The output data set produced by PROC MIRP gives, for each stock-keeping location, the inventory control parameter along with estimates of measures such as order quantities, backlogs, and inventory costs.

Chapter 1
Introduction to Inventory Replenishment Planning

Chapter Contents

Chapter 1
Introduction to Inventory Replenishment Planning

Overview

SAS Inventory Replenishment Planning uses historical demand data to determine inventory replenishment policies for single location or multi-echelon distribution systems. In addition to the demand data, it can use the cost of replenishment, the cost of holding inventory, the cost of backordering (stockouts), and target service levels to drive the policy identification. These inventory replenishment policies perform better than the standard EOQ (economic order quantity) policies that do not account for variation in customer demand and replenishment order lead times.

Impact of Inventory

In most industries, inventory is the foundation of conducting business. Consider the manufacturing industry, where it is necessary to coordinate both inventory-producing and inventory-consuming activities. There are inventories for multiple processing stages at multiple locations in the course of turning raw materials into components, producing spare parts, and ultimately creating finished goods.

In the retail industry, companies maintain large volumes of different items at various locations. They must monitor quantities, estimate usage, and place orders for replenishment. Slow-moving items are discontinued, while new items are introduced.

In the service industry, inventories are critical in providing the services that customers require. For instance, where would the hospital industry be without adequate supplies of surgical instruments and medicines? And how would a major package delivery company function without an inventory of trucks and spare parts?

Inventories are the lifeblood of a business and are essential to running it efficiently and profitably. When doing business with a company, customers often will not tolerate product unavailability or delays in delivery. In some cases, a shortage may be only a small inconvenience (such as selecting a different video at the rental store), while sometimes it may cause a severe problem (such as interrupting production-line activity at a computer manufacturer). On some occasions, sporadic shortages can be expected, but frequent shortages may ultimately erode a company's reputation and reduce their market share.

Inversely, overabundant, slow-moving inventories can place a serious strain on a company's available capital and the company's ability to take advantage of financial opportunities. Frequent shortages or excessive inventories are telltale signs of a company headed in the wrong direction. Zipkin (2000) notes that "we have understood for some time, at least in principle, that sound, careful inventory management is critical to a firm's strategic viability."

The scope of inventory-dependent operations is tremendous. In March 2002, U.S. businesses alone maintained about $1.117 trillion worth of inventories, or roughly 1.38 times their total monthly sales. Thus, effective management of inventory can have a big impact on profitability. Recently, much success has come to retailers that focus their operations on keeping their inventories lean. Less has become more, and intelligent inventory replenishment planning is a major key toward realizing that goal. In order to compete effectively in today's business world, it is imperative that adequate inventories are maintained efficiently.

Function of Inventory

Zipkin (2000) states that "most of the important functions of inventories can be understood in terms of the various types of mismatches that arise between supply and demand processes." Typically, these processes cannot be matched perfectly. As a result, inventory acts as the buffer between them to reduce the effect of their incompatibilities. As expected, conflicts often arise. The most common of these is a shortage—the failure to meet demand when it occurs. Thus, one primary function of inventory is to prevent or limit shortages. Consider some of the characteristics of supply and demand processes that Zipkin (2000) suggests can contribute to this inherent incongruity:

Supply:

- economies of scale (production and delivery)
- capacity limits (production and delivery)
- delays in replenishment (order lead time)

Demand:

- steady or intermittent demand
- variations in demand over time (trend, seasonality)
- unpredictable demand variations (random)

Each of these factors can contribute to uncertainty. When dealing with uncertainty, the traditional objective of inventory control models is to minimize expected costs. Consider some of the costs associated with most inventory control systems:

Inventory-Related Costs:

- replenishment cost (or fixed ordering cost)
  - cost of processing orders
  - cost is independent of replenishment quantity
- inventory holding cost
  - opportunity cost of capital invested in inventory
  - warehousing cost
  - handling and counting costs
  - other costs such as insurance and taxes
- stockout cost
  - cost of backordering
  - penalty cost for lost sales

When ordering, holding, and stockout costs are all known, SAS Inventory Replenishment Planning can be used to calculate optimal inventory replenishment policies for a single location. However, estimating stockout costs can be difficult, so a service level requirement is often substituted. In this case, a heuristic algorithm can be used to calculate nearly optimal policies, subject to requirements based on a choice of several different service measures.

Zipkin (2000) states that "the distinction between predictable (or deterministic) and unpredictable (or stochastic) processes is perhaps the single most significant dividing line between different [inventory] systems." SAS Inventory Replenishment Planning takes into account this unpredictability (or uncertainty) when calculating inventory replenishment policies for use in inventory management.

Summary of Functionality

SAS Inventory Replenishment Planning provides essential aid to decision making in inventory management by answering two fundamental questions:

- When should orders be placed to restock inventory?
- How much should be ordered?

The IRP and MIRP procedures in SAS Inventory Replenishment Planning provide the ability to transform raw demand transaction data and order lead time estimates into rules for managing product inventory levels.

Using estimates of review-time demand and replenishment order lead time along with the associated inventory costs for ordering, holding, and stockouts, the IRP procedure calculates optimal (s, S) or (s, nQ) policies for single-location or two-echelon inventory distribution systems. If the stockout penalty cost is unknown, one of several service measures can be substituted and the IRP procedure can calculate nearly optimal (s, S) or (s, nQ) policies. In both cases, PROC IRP provides an estimate of service measures for the purpose of evaluating projected policy performance. For details, see Chapter 2, "The IRP Procedure."

For multi-echelon supply chains, the MIRP procedure calculates optimal inventory investment decisions using base-stock replenishment policies and ready rate service level requirements. In addition, PROC MIRP provides estimates of measures such as order quantities, backlogs, and inventory costs. For details, see Chapter 3, "The MIRP Procedure."

References

Graves, S. C., Rinnooy Kan, A. H. G., and Zipkin, P. H., eds. (1993), *Handbooks in Operations Research and Management Science, 4: Logistics of Production and Inventory*, Netherlands: North-Holland.

Zipkin, P. H. (2000), *Foundations of Inventory Management*, New York: McGraw-Hill.

Chapter 2
The IRP Procedure

Chapter Contents

Chapter 2
The IRP Procedure

Overview

The IRP procedure provides the ability to calculate periodic-review inventory replenishment policies for single-location and two-echelon distribution inventory systems. These policies are determined through a number of algorithms that are controlled by user-specified options.

PROC IRP can calculate two types of replenishment policies, (s, S) policies and (s, nQ) policies. For details, see the "Replenishment Policies" section on page 33.

An *optimal* policy is defined as a policy that minimizes the average cost — the total of ordering, holding, and backorder penalty costs. PROC IRP uses several heuristic algorithms to approximate optimal policies to meet the user-specified service constraints. If the penalty cost information is available, PROC IRP can also calculate optimal (s, S) or (s, nQ) policies for single-location inventory systems.

Getting Started

Single-Location Inventory Systems

In a single-location inventory system, customers (or demand transactions) request a random amount of an item (SKU). Customer orders are filled from on-hand inventory. If insufficient inventory is available, the order is filled partially with available inventory and any unsatisfied portion is backlogged (or backordered). The *inventory position*, which is on-hand inventory plus inventory-on-order minus backorders, is monitored periodically. Based on the current inventory position, the replenishment policy will determine whether or not a replenishment order should be placed from an outside supplier.

Periodic review is the most common type of review process. Inventory is counted or evaluated periodically (say, monthly) at discrete points in time to determine if a replenishment order needs to be placed. Replenishment decisions can be made only at those points. The time between two review points is called the *review period*.

When a replenishment order is placed, there may be a delay between when the order is placed and when the order arrives. This delay is called the *lead time*, and is specified in the same units as the review period. For example, if the review period is one day (i.e., inventory is reviewed daily) and the lead time is one week, the lead time would be specified as seven days. The IRP procedure accounts for demand that occurs during the lead time.

The size of the demand that occurs during one review period is called the *review-time demand*. When demand is stationary (that is, demand stays relatively constant across review periods), PROC IRP requires only the mean and variance of review-time demand. For example, these values may be estimates calculated using a forecast engine prior to invoking PROC IRP. When demand is nonstationary, information must be provided to PROC IRP about the lead-time demand rather than the review-time demand; see the "LEADTIMEDEMAND Statement" section on page 23 for more information about lead-time demand.

PROC IRP calculates inventory replenishment policies using this information — inventory position, lead time, and review-time demand — together with user-specified inventory-related costs and policy restrictions.

As a simple example, consider a single store that carries five different items (SKUs), which are ordered from an outside supplier. Calculation of demand forecasts and inventory review is done weekly. The manager wants to calculate (s, S) policies that will minimize expected holding and ordering costs and achieve a target fill rate of 95%. Table 2.1 summarizes the demand, lead time, and cost information for these items. Note that the lead times are expressed in terms of weeks (either one, two, or three weeks), because the review period is one week.

Table 2.1. Data Summary

Sku	Holding Cost	Ordering Cost	Lead Time	Mean of Demand	Variance of Demand
A	0.35	90	1	125.1	2170.8
B	0.05	50	2	140.3	1667.7
C	0.12	50	3	116.0	3213.4
D	0.10	75	1	291.8	5212.4
E	0.45	75	2	134.5	1980.5

This information is stored in a data set called skuInfo and displayed in Figure 2.1. The mean and variance of one-period demand are given by the RTDmean and RTDvar variables. The lead time is fixed (that is, it has zero variance) and is given by the LTmean variable. Similarly, holding and ordering costs are given by the holdingCost and fixedCost variables. Finally, the serviceLevel variable specifies the desired service level.

```
                                Input Data Set

                holding     fixed                                           service
  Obs     sku     Cost       Cost     LTmean     RTDmean     RTDvar      Level

   1       A      0.35        90         1        125.1      2170.8       0.95
   2       B      0.05        50         2        140.3      1667.7       0.95
   3       C      0.12        50         3        116.0      3213.4       0.95
   4       D      0.10        75         1        291.8      5212.4       0.95
   5       E      0.45        75         2        134.5      1980.5       0.95
```

Figure 2.1. Input Data Set skuInfo

The following IRP procedure call can be used to calculate the inventory policies.

```
proc irp data=skuInfo out=policy;
   itemid sku;
   holdingcost holdingCost;
   leadtime / mean=LTmean;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel;
run;
```

The REVIEWTIMEDEMAND statement specifies the variables that contain the mean and variance of review-time demand. Similarly the LEADTIME statement identifies the variable containing the lead time, and the SERVICE statement identifies the variable that specifies the desired service levels. Fill rate is the default service measure and (s, S) policies are the default policy type, so no extra options or statements are needed. The variables RTDmean, RTDvar, LTmean, fixedCost, holdingCost, and serviceLevel are all default variable names, so you do not need to specify them in any statements. Thus, the following IRP procedure call would have produced the same results:

```
proc irp data=skuInfo out=policy;
   itemid sku;
run;
```

The output data set policy is displayed in Figure 2.2.

Policy Data Set

Obs	sku	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost	inventory Ratio
1	A	223	474	132.443	6.2118	0.38748	81.2284	1.05870
2	B	359	866	231.793	7.1533	0.24066	23.6226	1.65213
3	C	483	805	215.502	6.1992	0.29468	40.5943	1.85777
4	D	468	1109	281.858	14.5779	0.36662	55.6826	0.96593
5	E	390	604	127.930	6.7485	0.46591	92.5115	0.95115

Obs	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	0.049654	0.94456	0.95046	0.86962	FR-SS-NO	SUCCESSFUL
2	0.050986	0.60528	0.94968	0.88107	FR-SS-NO	SUCCESSFUL
3	0.053441	0.53828	0.94986	0.90762	FR-SS-NO	SUCCESSFUL
4	0.049958	1.03527	0.95006	0.84650	FR-SS-NO	SUCCESSFUL
5	0.050175	1.05136	0.95019	0.86847	FR-SS-NO	SUCCESSFUL

Figure 2.2. Policy Output Data Set

The reorderLevel variable gives the reorder level, s, and the orderUpToLevel variable gives the order-up-to level, S. For example, for sku A, any time the inventory position is observed to be less than or equal to 223 at a review point, a replenishment order is placed to bring the inventory position up to 474. The _status_ variable indicates that the optimization was successful for all observations. The _algorithm_ variable gives information about the algorithm used; namely, a fill rate ('FR') service level heuristic was used to calculate (s, S) policies ('SS'), using a normal distribution ('NO') for lead-time demand and (lead time + review time)-demand. The remaining variables report several estimated inventory metrics to evaluate the performance of the underlying policy (see the "OUT= Data Set" section on page 30 for information about these variables).

Two-Echelon Distribution Inventory Systems

A two-echelon distribution inventory system consists of a single warehouse and multiple retail locations. The retail locations do not incur a fixed cost when ordering from the warehouse and thus follow a base-stock policy. The warehouse, however, incurs a fixed cost when ordering from an outside supplier and thus can follow an (s, S) or (s, nQ) policy. PROC IRP can find nearly optimal policies for two-echelon distribution inventory systems with different service constraints on the retail locations.

Consider a warehouse-retailer distribution problem with two items. For sku A, the warehouse is in Raleigh, NC, and the retail locations are located in Atlanta, GA, Baltimore, MD, and Charleston, SC. For sku B, the warehouse is in Greensboro, NC, and the retail locations are in Atlanta, GA, and Charleston, SC. The demand, lead time, and cost information of each item is stored in a data set called skuInfo2, as shown in Figure 2.3.

Input Data Set

Obs	sku	warehouse	location	holdingCost	fixedCost	LTmean	RTDmean	RTDvar	serviceLevel
1	A	Raleigh, NC		0.35	90	1	125.1	2170.8	.
2	A	Raleigh, NC	Atlanta, GA	0.70	.	2	32.6	460.2	0.95
3	A	Raleigh, NC	Baltimore, MD	0.70	.	2	61.8	1133.5	0.95
4	A	Raleigh, NC	Charleston, SC	0.70	.	1	30.7	577.1	0.95
5	B	Greensboro, NC		0.05	50	2	140.3	1667.7	.
6	B	Greensboro, NC	Atlanta, GA	0.10	.	2	68.4	907.3	0.95
7	B	Greensboro, NC	Charleston, SC	0.10	.	1	71.9	760.4	0.95

Figure 2.3. Input Data Set skuInfo2

The location and serviceLevel variables have missing values when the observation corresponds to a warehouse. PROC IRP treats the current observation as a warehouse if the corresponding entry for the location variable is missing. Similarly, the fixedCost variable has missing values for the retail locations since the retail locations follow base-stock policies and do not incur ordering costs. Only the warehouses incur ordering costs because they replenish from an outside supplier.

The following IRP procedure call can be used to calculate inventory policies for the warehouses and the retail locations.

```
proc irp data=skuInfo2 out=policy2;
   itemid sku warehouse;
   location location;
   holdingcost holdingCost;
   leadtime / mean=LTmean;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel;
run;
```

The output data set policy2 is displayed in Figure 2.4. The reorderLevel variable gives the reorder level, s, and the orderUpToLevel variable gives the order-up-to level S. Note that for the retailers, the order-up-to level is one greater than the reorder level, since the retailers follow base-stock policies. The _status_ variable indicates that the optimization was successful for all observations. The _algorithm_ variable gives information about the algorithm used; namely, a fill rate ('FR') service level was used for the retailers, the warehouses follow (s, S) ('SS') policies and the retailers follow base-stock ('BS') policies, and the gamma distribution ('GA') was used for lead-time demand and (lead time + review time)-demand for all locations. The remaining variables report several estimated inventory metrics to evaluate the performance of the underlying policy (see the "OUT= Data Set" section on page 30 for information about these variables).

Policy Data Set

Obs	sku	warehouse	location	reorderLevel	orderUpToLevel	avgInventory	avgBackorder	avgOrderFreq	avgCost
1	A	Raleigh, NC		124	376	58.836	31.3840	0.38748	218.928
2	A	Raleigh, NC	Atlanta, GA	170	171	66.996	1.9506	0.99984	46.897
3	A	Raleigh, NC	Baltimore, MD	296	297	99.730	3.5886	1.00000	69.811
4	A	Raleigh, NC	Charleston, SC	133	134	66.792	1.8710	0.99820	46.754
5	B	Greensboro, NC		256	763	149.251	27.6111	0.24066	34.560
6	B	Greensboro, NC	Atlanta, GA	299	300	84.929	3.7109	1.00000	8.493
7	B	Greensboro, NC	Charleston, SC	219	220	65.717	3.7938	1.00000	6.572

Obs	inventoryRatio	backorderRatio	turnover	fillRate	readyRate	_algorithm_	_status_
1	0.47031	0.25087	2.12625	0.76569	0.60867	__-SS-GA	SUCCESSFUL
2	2.05509	0.05983	0.48660	0.94982	0.92836	FR-BS-GA	SUCCESSFUL
3	1.61375	0.05807	0.61967	0.94890	0.91534	FR-BS-GA	SUCCESSFUL
4	2.17562	0.06094	0.45964	0.94919	0.93515	FR-BS-GA	SUCCESSFUL
5	1.06380	0.19680	0.94003	0.82337	0.71705	__-SS-GA	SUCCESSFUL
6	1.24165	0.05425	0.80538	0.94995	0.90168	FR-BS-GA	SUCCESSFUL
7	0.91400	0.05277	1.09409	0.95019	0.88519	FR-BS-GA	SUCCESSFUL

Figure 2.4. Output Data Set policy2

Syntax

The following statements are used in PROC IRP:

PROC IRP *options* **;**
  HOLDINGCOST *variable* **;**
  ITEMID *variables* **;**
  LEADTIME / *lead time options* **;**
  LEADTIMEDEMAND / *lead-time demand options* **;**
  LOCATION *variable* / *location options* **;**
  PENALTY *variable* / *penalty options* **;**
  POLICYTYPE *variable* **;**
  REPLENISHMENT / *replenishment options* **;**
  REVIEWTIMEDEMAND / *review-time demand options* **;**
  SERVICE / *service options* **;**

Functional Summary

The following tables outline the options available for the IRP procedure classified by function.

Table 2.2. Constraints and Policy Specifications

Description	Statement	Option
maximum ordering frequency *variable*	REPLENISHMENT	MAXFREQ=
minimum order size *variable*	REPLENISHMENT	MINSIZE=
base lot size *variable*	REPLENISHMENT	LOTSIZE=
policy type *variable*	POLICYTYPE	
service type *variable*	SERVICE	TYPE=
service level *variable*	SERVICE	LEVEL=

Table 2.3. Cost Specifications

Description	Statement	Option
fixed cost *variable*	REPLENISHMENT	FCOST=
holding cost *variable*	HOLDINGCOST	
penalty cost *variable*	PENALTY	COST=

Table 2.4. Data Set Specifications

Description	Statement	Option
input data set	PROC IRP	DATA=
output data set	PROC IRP	OUT=

Table 2.5. Identifier Variables

Description	Statement	Option
item id *variables*	ITEMID	
location *variable*	LOCATION	

Table 2.6. Lead Time Specifications

Description	Statement	Option
lead time mean *variable*	LEADTIME	MEAN=
lead time variance *variable*	LEADTIME	VARIANCE=
maximum allowed value of coefficient of variation for lead time	LEADTIME	MAXCOV=

Table 2.7. Lead-Time Demand Specifications

Description	Statement	Option
lead-time demand mean *variable*	LEADTIMEDEMAND	MEAN=
lead-time demand variance *variable*	LEADTIMEDEMAND	VARIANCE=
maximum allowed value of coefficient of variation for lead-time demand	LEADTIMEDEMAND	MAXCOV=

Table 2.8. Miscellaneous Options

Description	Statement	Option
maximum number of items for which input error messages are printed	PROC IRP	MAXMESSAGES=
estimate of the maximum number of retail locations	LOCATION	NLOCATIONS=

Table 2.9. Optimization Control Specifications

Description	Statement	Option
maximum number of iterations	PROC IRP	MAXITER=
type of policy algorithm	PROC IRP	METHOD=
specifies calculation of optimal policies	PENALTY	OPTIMAL
controls the scaling of demand and cost parameters	PENALTY	SCALE=
criterion to determine $S - s$ or Q	REPLENISHMENT	DELTA=

Table 2.10. Review-Time Demand Specifications

Description	Statement	Option
review-time demand mean *variable*	REVIEWTIMEDEMAND	MEAN=
review-time demand variance *variable*	REVIEWTIMEDEMAND	VARIANCE=
maximum allowed value of coefficient of variation for review-time demand	REVIEWTIMEDEMAND	MAXCOV=

PROC IRP Statement

PROC IRP *options* **;**

The following options can appear in the PROC IRP statement.

DATA=*SAS-data-set*
names the SAS data set that contains information about the items to be analyzed. Required information includes the mean and variance of review-time demand, mean replenishment order lead time, per unit holding cost, fixed replenishment cost, and the target service level or backorder penalty cost. Optional information may be supplied with other variables for use by the procedure. For single-location systems, every observation corresponds to an individual inventory item to be analyzed. For two-echelon distribution systems, every observation corresponds to an inventory item-location pair, and these pairs must be grouped together by item.

The DATA= input data set must be sorted by the variables specified with the ITEMID statement. See the "Input Data Set" section on page 28 for more information about the variables in this data set. If the DATA= option is omitted, the most recently created SAS data set is used.

MAXITER=*maxiter*
specifies the maximum number of iterations permitted for the heuristic algorithm to calculate inventory replenishment policies. The default value of *maxiter* is 100. This option is ignored when the OPTIMAL option is specified on the PENALTY statement.

MAXMESSAGES=*maxmessages*
MAXMSG=*maxmessages*
specifies the maximum number of different items in the DATA= input data set for which input error messages are printed to the SAS log. The default value of *maxmessages* is 100.

METHOD= SERVICE | PENALTY
specifies the algorithm used for calculating the inventory replenishment policies. If METHOD= is specified as PENALTY, PROC IRP uses backorder penalty costs to determine the replenishment policy. If METHOD= is SERVICE, then service level requirements are used to calculate the replenishment policy. The default value of METHOD= is SERVICE.

OUT=*SAS-data-set*
specifies a name for the output data set that contains inventory replenishment policies, service measures estimates, and other inventory metrics as determined by PROC IRP. This data set also contains all of the variables specified with the ITEMID statement. Every observation in the DATA= input data set has a corresponding observation in this output data set. See the "OUT= Data Set" section on page 30 for information about the variables in this data set. If the OUT= option is omitted, the SAS system creates a data set and names it according to the DATA*n* naming convention.

HOLDINGCOST Statement

HOLDINGCOST *variable* **;**
HCOST *variable* **;**

The HOLDINGCOST statement identifies the variable in the DATA= input data set that specifies the per-period per-unit holding cost of each item. Negative, zero, and missing values are not permitted. If this statement is not specified, PROC IRP looks for a default variable named HOLDINGCOST. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

ITEMID Statement

ITEMID *variables* **;**
ID *variables* **;**
SKUID *variables* **;**

The ITEMID statement specifies the variables in the DATA= input data set that identify individual inventory items. For a single-location system, the ITEMID variables are primarily used to identify unique items in the input data set. However, each observation is processed independently, regardless of whether or not the values of the ITEMID variables are unique. Thus, you can include any variables that may not necessarily pertain to the descriptions of the items in the list. All the variables specified by this statement are included in the output data set. Therefore, in addition to identifying inventory information (like SKU), the ITEMID statement can also be used in a single-location system to specify variables that will be carried through from the input data set to the output data set. See Example 2.1 on page 44 for an illustration.

For a two-echelon system, the ITEMID statement specifies the variables in the DATA= input data set that are used in grouping the observations in the input data set. Each group identifies a single item that is shipped from a warehouse to one or more retailers; each individual observation within a group corresponds to a single warehouse or retailer. The observations within a group are used together to process the group. As in the single-location case, the variables specified by the ITEMID statement are included in the output data set; however, in this case, the variables are used to process observations in groups rather than independently. Thus, the ITEMID statement cannot be used (as in the single-location system) to simply copy variables from the input data set to the output data set. Rather, a simple data step can be performed after a call to PROC IRP to merge variables from the input and output data sets.

If the ITEMID statement is not specified, PROC IRP halts with an error. Furthermore, PROC IRP expects the DATA= input data set to be sorted by the variables specified by the ITEMID statement. The ITEMID statement behaves much like the BY statement; therefore, you can use options such as DESCENDING and NOTSORTED on the ITEMID statement. Refer to SAS System documentation for more information on the BY statement.

LEADTIME Statement

LEADTIME / *lead time options* **;**
LTIME / *lead time options* **;**

The LEADTIME statement identifies the variables in the DATA= input data set that contain the mean and variance of the replenishment order lead time. This information is used to calculate the mean and variance of lead-time demand. The replenishment order lead time should be specified using the same scale as the review periods. This statement is ignored if the LEADTIMEDEMAND statement is specified.

MEAN=*variable*
identifies the variable in the DATA= input data set that contains the mean of the replenishment order lead time. Negative, zero, and missing values are not permitted. If this option is omitted, PROC IRP looks for a default variable named LTMEAN. If this variable is not found in the DATA= data set, PROC IRP halts with an error.

VARIANCE=*variable*
VAR=*variable*
identifies the variable in the DATA= input data set that contains the variance of the replenishment order lead time. Negative and missing values are interpreted as 0. If this option is omitted, a value of 0 will be used for all observations.

MAXCOV=*maxcov*
specifies the maximum allowed value of the coefficient of variation for replenishment order lead time. Items with coefficient of variation (ratio of the standard deviation and mean) of lead time greater than *maxcov* are not processed. The default value of *maxcov* is 10.

LEADTIMEDEMAND Statement

LEADTIMEDEMAND / *lead-time demand options* **;**
LTDEMAND / *lead-time demand options* **;**

The LEADTIMEDEMAND statement identifies the variables in the DATA= input data set that contain the mean and variance of lead-time demand (that is, the amount of demand that occurs during the lead time). The IRP procedure uses the review-time demand and lead time information to calculate the parameters of lead-time demand. Instead of specifying the parameters of lead time, you can directly specify the mean and variance of lead-time demand with the LEADTIMEDEMAND statement. This feature is especially useful if lead time is greater than review time and demand is not stationary.

If this statement is specified, both the MEAN= and VARIANCE= options must be specified, and the LEADTIME statement is ignored. Since the inventory is periodically reviewed, the lead time in consideration should start after one review period. See Example 2.4 on page 53 for an illustration.

MEAN=*variable*
identifies the variable in the DATA= input data set that contains the mean of the demand during lead time. Negative, zero, and missing values are not permitted.

VARIANCE=*variable*
VAR=*variable*
identifies the variable in the DATA= input data set that contains the variance of the demand during lead time. Negative, zero, and missing values are not permitted.

MAXCOV=*maxcov*
specifies the maximum allowed value of the coefficient of variation for lead-time demand. Items with coefficient of variation (ratio of the standard deviation and mean) of lead-time demand greater than *maxcov* are not processed. The default value of *maxcov* is 10.

LOCATION Statement

LOCATION *variable / location options* **;**
LOC *variable / location options* **;**

The LOCATION statement identifies the character variable in the DATA= data set that identifies the retail locations for the two-echelon distribution inventory problem. The value of the LOCATION variable should be missing if the current observation corresponds to a warehouse. This statement is required to solve two-echelon distribution inventory problems. If this statement is omitted, each observation is treated as a separate single-location inventory problem.

NLOCATIONS=*nlocations*
NLOCS=*nlocations*
specifies an estimate of the maximum number of retail locations in a single item group for the two-echelon distribution inventory problem. This option is used for initial memory allocation. The default value is 50.

PENALTY Statement

PENALTY */ penalty options* **;**

The PENALTY statement enables you to specify backorder penalty cost information. This statement is ignored if the METHOD= option is specified as SERVICE.

COST=*variable*
identifies the variable in the DATA= input data set that specifies the per period per unit item penalty cost for backlogged demand. Negative, zero, and missing values are not permitted. The value of this variable must also be greater than or equal to 1.5 times the value of the HOLDINGCOST variable. This limitation is to avoid accidental user input errors and to guarantee a minimum ready rate of at least 60 percent. If the METHOD= option is specified as PENALTY and this option is not specified, PROC IRP looks for a default variable named PENALTYCOST. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

OPTIMAL
OPT

specifies that an optimal policy will be calculated. This option is valid only if the LOCATION statement is not specified. By default, PROC IRP uses a heuristic method to calculate nearly optimal policies. See the "OPTIMAL Option" section on page 39 for more information.

SCALE=*scale*

controls the initial scaling of demand and cost parameters for optimal policy calculations. Initial scaling takes place if the calculated mean of (lead time + review time)-demand is greater than *scale*. This option is ignored if the OPTIMAL option is not specified. Valid values are between 50 and 10,000, and the default value is 100. In general, the default scaling is sufficient to produce fast and accurate results. If desired, more accuracy may be obtained at the expense of longer execution time by increasing *scale* (thus decreasing the effective scaling). However, increasing *scale* increases the demand on memory and may result in an error. See the "OPTIMAL Option" section on page 39 for more information.

POLICYTYPE Statement

POLICYTYPE *variable* **;**
PTYPE *variable* **;**

The POLICYTYPE statement identifies the variable in the DATA= input data set that specifies the type of inventory replenishment policy to be calculated. The values allowed for the variable specified in the POLICYTYPE statement are listed in Table 2.11. See the "Replenishment Policies" section on page 33 for more information on policy types.

Table 2.11. Valid Values for the POLICYTYPE Variable

Value	Policy Type
BS	base-stock policy
SS	(s, S) policy (default)
NQ	(s, nQ) policy, fixed ordering cost for each lot ordered
RQ	(s, nQ) policy, single fixed ordering cost independent of the number of lots ordered

If this statement is not specified, the default value, SS, is used.

REPLENISHMENT Statement

REPLENISHMENT / *replenishment options* **;**
ORDER / *replenishment options* **;**
REP / *replenishment options* **;**

DELTA= POWER | EOQ
specifies the method used for calculating the difference, $\Delta = S - s$, for (s, S) policies or the base lot size, Q, for (s, nQ) policies. Valid values of DELTA= are POWER and EOQ. The default value is POWER. See the "Policy Algorithm" section on page 36 for more information.

FCOST=*variable*
identifies the variable in the DATA= input data set that specifies the fixed ordering cost of placing a replenishment order. Negative and missing values are interpreted as 0. If this option is not specified, PROC IRP looks for a default variable named FIXEDCOST. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

LOTSIZE=*variable*
identifies the variable in the DATA= input data set that specifies the prespecified lot size (Q) for a replenishment order. Replenishment orders placed can only be made in multiples of the value of this variable. Negative, zero, and missing values are ignored.

MINSIZE=*variable*
identifies the variable in the DATA= input data set that contains the minimum allowable replenishment order size. Negative and missing values are ignored, with the exception of -1. A value of -1 is a special flag and sets the minimum order size to 1.5 times the average one-period demand. If this statement is omitted, a value of 0 will be used for all observations.

MAXFREQ=*variable*
identifies the variable in the DATA= input data set that contains the maximum allowable average ordering frequency. In practice, the fixed cost of placing an order can be difficult to estimate; therefore, this variable enables the user to put a limit on the frequency with which orders are placed. Negative, zero, and missing values are ignored.

REVIEWTIMEDEMAND Statement

REVIEWTIMEDEMAND / *review-time demand options* **;**
RTDEMAND / *review-time demand options* **;**

The REVIEWTIMEDEMAND statement identifies the variables in the DATA= input data set that contain the mean and variance of the review-time demand. When using the REVIEWTIMEDEMAND statement, demand over the review periods is assumed to be stationary and independent.

MEAN=*variable*

identifies the variable in the DATA= input data set that contains the mean of the demand during a single inventory review period. Missing values and values less than 1 are not permitted. However, the mean of review-time demand at the warehouse (in the two-echelon distribution problem) can be set to missing to instruct PROC IRP to automatically calculate the mean and variance of demand at the warehouse as the sum of the means and variances of demand at the retail locations. If this option is omitted, PROC IRP looks for a default variable named RTDMEAN. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

VARIANCE=*variable*
VAR=*variable*

identifies the variable in the DATA= input data set that contains the variance of the demand during a single inventory review period. Negative and missing values are interpreted as 0. If this statement is omitted, PROC IRP looks for a default variable named RTDVAR. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

MAXCOV=*maxcov*

specifies the maximum allowed value of the coefficient of variation for review-time demand. Items with coefficient of variation (ratio of the standard deviation and mean) of review-time demand greater than *maxcov* are not processed. The default value of *maxcov* is 10.

SERVICE Statement

SERVICE / *service options* **;**

The SERVICE statement identifies the variables in the DATA= input data set that specify the type and the desired level of the service measure to be used by the inventory policy algorithm. This statement is ignored if the METHOD= option on the PROC IRP statement is specified as PENALTY.

LEVEL=*variable*

identifies the variable in the DATA= input data set that specifies the desired service level for the service measure specified with the TYPE= option. Common ranges of service level are $[0.80, 0.99]$ for fill rate and ready rate, and $[0.01, 0.20]$ for backorder ratio. Valid values for fill rate and ready rate are between 0.600 and 0.999 and for backorder ratio between 0.001 and 0.400. If the METHOD= option is specified as SERVICE and this option is not specified, PROC IRP looks for a default variable named SERVICELEVEL. If this variable is not found in the DATA= input data set, PROC IRP halts with an error.

TYPE=*variable*

identifies the variable in the DATA= input data set that specifies the type of service measure to be used by the inventory replenishment algorithm. Only one service measure per item can be specified in a single procedure invocation. The values allowed for the variable specified in the TYPE= option are listed in Table 2.12.

Table 2.12. Valid Values for the SERVICETYPE Variable

Value	Service Measure
FR	Fill rate (default)
RR	Ready rate
BR	Backorder ratio

If this option is not specified, the default value, FR, is used.

Details

This section provides detailed information about the use of the IRP procedure. The material is organized in subsections that describe different aspects of the procedure.

Input Data Set

PROC IRP uses data from the DATA= input data set with key variable names being used to identify the appropriate information. Table 2.13 lists all of the variables associated with the input data set and their interpretation by the IRP procedure. The variables are grouped according to the statement with which they are specified.

Table 2.13. PROC IRP Input Data Set and Associated Variables

Statement	Variable Name	Interpretation
HOLDINGCOST	HOLDINGCOST	holding cost
ITEMID	ITEMID	item identifier
LOCATION	LOCATION	retail location identifier
LEADTIME	MEAN	lead time mean
	VARIANCE	lead time variance
LEADTIMEDEMAND	MEAN	lead-time demand mean
	VARIANCE	lead-time demand variance
PENALTY	COST	backorder penalty cost
POLICYTYPE	POLICYTYPE	policy type
REPLENISHMENT	FCOST	fixed ordering cost
	LOTSIZE	base lot size
	MAXFREQ	maximum ordering frequency
	MINSIZE	minimum order size
REVIEWTIMEDEMAND	MEAN	review-time demand mean
	VARIANCE	review-time demand variance
SERVICE	LEVEL	desired service level
	TYPE	service measure type

Some variables have default names and do not need to be specified in any of the procedure statements. These variables are listed in Table 2.14.

Table 2.14. PROC IRP Input Data Set Default Variable Names

Statement	Variable Name	Default Variable Name
HOLDINGCOST	HOLDINGCOST	HOLDINGCOST
LEADTIME	MEAN	LTMEAN
PENALTY	COST	PENALTYCOST
REPLENISHMENT	FCOST	FIXEDCOST
REVIEWTIMEDEMAND	MEAN VARIANCE	RTDMEAN RTDVAR
SERVICE	LEVEL	SERVICELEVEL

Missing Values in the Input Data Set

Table 2.15 summarizes the treatment of missing values for variables in the DATA= input data set.

Table 2.15. Treatment of Missing Values in the IRP Procedure

Statement	Variable Name	Value / Action Taken
HOLDINGCOST	HOLDINGCOST	input error: procedure moves to processing of next ITEMID group
LOCATION	LOCATION	current observation defines a warehouse
LEADTIME	MEAN VARIANCE	input error: procedure moves to processing of next ITEMID group 0
LEADTIMEDEMAND	MEAN VARIANCE	input error: procedure moves to processing of next ITEMID group input error: procedure moves to processing of next ITEMID group
PENALTY	COST	input error: procedure moves to processing of next ITEMID group if METHOD=PENALTY, otherwise ignored
POLICYTYPE	POLICYTYPE	'SS'
REPLENISHMENT	FCOST LOTSIZE MAXFREQ MINSIZE	0 value ignored value ignored 0
REVIEWTIMEDEMAND	MEAN VARIANCE	input error (unless the value of the LOCATION variable is also missing): procedure moves to processing of next ITEMID group 0 (or the sum of other ITEMID group values if the value of the LOCATION variable is missing)
SERVICE	LEVEL TYPE	input error: procedure moves to processing of next ITEMID group if METHOD=SERVICE, otherwise ignored 'FR'

OUT= Data Set

The OUT= data set contains the inventory replenishment policies for the items identified in the DATA= input data set. There is one observation for each observation in the DATA= input data set. If an error is encountered while processing an observation, information about the error is written to the OUT= data set.

Definitions of Variables in the OUT= Data Set

Each observation in the OUT= data set is associated with an individual inventory item (or SKU). The variables specified with the ITEMID statement will be copied to the OUT= data set. The following variables will also be added to the OUT= data set:

AVGBACKORDER
contains the estimated average backorders for the calculated inventory replenishment policy. Average backorders is the average amount of cumulative backorders in a review period.

AVGCOST
contains the estimated average cost per period for the calculated inventory replenishment policy. Average cost is the average cost (including holding, ordering and backorder penalty costs) incurred per review period.

AVGINVENTORY
contains the estimated average inventory for the calculated inventory replenishment policy. Average inventory is the average on-hand inventory at the end of a review period.

AVGORDERFREQ
contains the estimated average ordering frequency for the calculated inventory replenishment policy. Average ordering frequency is the average number of replenishment orders placed per review period.

BACKORDERRATIO
contains the estimated backorder ratio for the calculated inventory replenishment policy. Backorder ratio is equal to average backorders divided by average demand.

FILLRATE
contains the estimated fill rate for the calculated inventory replenishment policy. Fill rate is the fraction of demand that is satisfied from on-hand inventory. If the OPTIMAL option is specified on the PENALTY statement, the FILLRATE variable will not be added to the OUT= data set.

INVENTORYRATIO
contains the estimated inventory ratio for the calculated inventory replenishment policy. Inventory ratio is equal to the average inventory divided by average demand.

ORDERUPTOLEVEL
specifies the order-up-to level, S, for (s, S) policies or the sum of the reorder level and the base lot size, $s + Q$, for (s, nQ) policies.

READYRATE

contains the estimated ready rate for the calculated inventory replenishment policy. Ready rate is the probability of no stockout in a review time period.

REORDERLEVEL

specifies the reorder level, s. The reorder level is the inventory level at which a replenishment order should be placed.

TURNOVER

contains the estimated turnover for the calculated inventory replenishment policy. Turnover is equal to the average demand divided by average inventory. The value of this variable will be set to missing if the estimated average inventory is 0.

ALGORITHM

indicates which algorithm was used to calculate the inventory replenishment policy. The value of the _ALGORITHM_ variable is in the form of XX-YY-ZZ, where XX indicates the type of optimization used, YY indicates type of policy calculated, and ZZ indicates the approximation used for both lead-time demand and (lead time + review time)-demand distributions. Possible values for the _ALGORITHM_ variable are specified in Table 2.16.

Table 2.16. Possible Values of the _ALGORITHM_ Variable

String	Value	Description
XX	PC	Penalty cost
	FR	Fill rate
	RR	Ready rate
	BR	Backorder ratio
YY	BS	$(S-1, S)$ base-stock policy
	SS	(s, S) policy (or (s, nQ, S) policy if a base lot size Q is specified)
	MS	modified (s, S) policy
	NQ	(s, nQ) policy, fixed ordering cost for each lot ordered
	MN	modified (s, nQ) policy, fixed ordering cost for each lot ordered
	RQ	(s, nQ) policy, single fixed ordering cost independent of the number of lots ordered
	MR	modified (s, nQ) policy, single fixed ordering cost independent of the number of lots ordered
ZZ	NO	Normal distribution
	GA	Gamma distribution

For additional information on modified policies listed in Table 2.16, see the "Modified Policies" section on page 39.

The ZZ portion of this variable will have a slightly different format when the OPTIMAL option is specified. See the "OPTIMAL Option" section on page 39 for details.

For two-echelon distribution systems, the XX portion of this variable will have value '__' when the current value of the LOCATION variable defines a warehouse (as no service constraints or penalty costs are applied at the warehouse).

SCALE

contains the value used to scale the demand and cost parameters during policy calculations. In the event that scaling is performed (the value of _SCALE_ is greater than 1), it should be noted that all values written to the OUT= data set are in original units. This variable is only added to the OUT= data set when the OPTIMAL option is specified on the PENALTY statement. For more information about scaling, see the "OPTIMAL Option" section on page 39.

STATUS

contains the completion status of the inventory replenishment algorithm. Possible values for the _STATUS_ variable are listed in Table 2.17.

Table 2.17. Possible Values of the _STATUS_ Variable

Value	Explanation
SUCCESSFUL	Successful completion
INVD_VALUE	Invalid value in the DATA= input data set
MAX_ITER	Maximum number of iterations reached
INSUF_MEM	Insufficient memory
BAD_DATA	Numerical or scaling problem encountered

Note that the estimates for average inventory and average backorders lose their accuracy if lead time is not an integer multiple of the review period, or if the variance of lead time is high.

Error Processing

For single location systems, PROC IRP processes each item (observation) individually. If an error occurs, PROC IRP stops processing the current item and writes information about the type of error to the _STATUS_ variable in the OUT= data set. Execution resumes with the next item.

For two-echelon distribution systems, PROC IRP processes items in groups (multiple observations) representing the warehouse and retail locations. If an error is detected for any of the corresponding observations, PROC IRP stops processing the current item group and the type of error is noted in the _STATUS_ variable for all items in the group. Execution resumes with the next item group.

At procedure termination, the value of the macro variable, _IRPIRP_, will be set appropriately to reflect the fact that errors were encountered during execution.

Macro Variable _IRPIRP_

PROC IRP defines a macro variable named _IRPIRP_ . This variable contains a character string that indicates the status of the procedure. It is set at procedure termination. The form of the _IRPIRP_ character string is STATUS=*status* NSUCCESS=*nsuccess* NFAIL=*nfail*, where *nsuccess* is the number of items successfully processed, *nfail* is the number of items for which the policy calculation has failed, and *status* can be one of the following:

- SUCCESSFUL (indicates successful completion of the procedure)
- RUNTIME_ERROR (indicates that policy calculations failed for at least one item or item group in the DATA= input data set)
- SYNTAX_ERROR (indicates failure due to a procedure syntax error)
- MEMORY_ERROR (indicates failure during procedure initialization or data input parsing due to insufficient memory)

This information can be used when PROC IRP is one step in a larger program that needs to determine whether the procedure terminated successfully or not. Because _IRPIRP_ is a standard SAS macro variable, it can be used in the ways that all macro variables can be used.

Replenishment Policies

PROC IRP calculates two types of replenishment policies:

(s, S) *Policy*: When the inventory position falls to or below the reorder level, s, an order is placed so as to bring the inventory position to the order-up-to level, S. In other words, if the inventory position is y, and $y \leq s$, then an order of size $S - y$ is placed. The (s, S) policy is sometimes referred to as the *min-max* policy. Note that the size of the replenishment order is always greater than or equal to $S - s$. When $S = s + 1$, the (s, S) policy is called a *base-stock policy* (also called an "order-up-to policy," "one-to-one replenishment policy," or "installation stock policy").

(s, nQ) *Policy*: When the inventory position falls to or below the reorder level, s, an order is placed to bring the inventory position just above s. The size of this order is a multiple of the base lot size, Q. In other words, if the inventory position is y, and $y \leq s$, then an order of size nQ is placed, where n is the smallest integer such that $y + nQ > s$. In this case, both s and Q are decision variables; you can use the LOTSIZE= option if Q is to be a prespecified value rather than a decision variable. Note that if $Q = 1$, the (s, nQ) policy becomes a base-stock policy.

Both (s, S) and (s, nQ) policies are special cases of a policy class called (s, nQ, S) policies. In this case, if the inventory position is y and $y \leq s$, an order of size nQ is placed where n is the smallest integer such that $y + nQ \geq S$. Usually the base lot size, Q, is specified by the supplier. Note that if $Q = 1$, the (s, nQ, S) policy becomes an (s, S) policy, and if $S = s + 1$, the (s, nQ, S) policy becomes an (s, nQ) policy.

For single-location inventory systems under standard assumptions (independent customer demands, full backordering of unfulfilled demand, fixed replenishment ordering costs, linear inventory holding costs, and linear backorder penalty costs), (s, nQ)

policies are known to be suboptimal and (s, S) policies are known to be optimal. Although (s, S) policies are optimal, the restricted order size under an (s, nQ) policy may better facilitate easy packaging, transportation, and coordination in some situations.

Inventory Costs

Since the objective of inventory planning is usually to minimize costs, the assumptions about the cost structure are important. There are three types of costs: ordering, holding, and penalty (backordering) costs.

Ordering cost is the cost incurred every time a replenishment order is placed. This fixed cost includes the expense associated with processing the order and is typically independent of the size of the order.

Holding cost is the cost of carrying inventory and may include the opportunity cost of money invested, the expenses incurred in running a warehouse, handling and counting costs, the costs of special storage requirements, deterioration of stock, damage, theft, obsolescence, insurance and taxes. The most common convention is to specify holding cost (per period per unit item) as a fixed percentage of the unit cost of the item. This cost is then applied to the average inventory.

Penalty (*backordering or shortage*) *cost* is the cost incurred when a stockout occurs. This cost may include the cost of emergency shipments, cost of substitution of a less profitable item, or cost of lost goodwill. For instance, will the customer ever return? Will the customer's colleagues be told of the disservice? The most common convention is to specify penalty cost as per period per unit item and then apply it to average backorders.

In practice, it is often difficult to estimate the ordering (replenishment) cost and the penalty cost. As a result, practitioners often put restrictions on the ordering frequency rather than estimating the cost of ordering. Likewise, specific target levels for service measures can be substituted for the penalty cost.

Service Measures

Service measures are often used to evaluate the effectiveness of an inventory replenishment policy. You can influence policy calculations by imposing desired service level requirements. PROC IRP supports the use of three different service constraints:

- *Fill Rate* — the fraction of demand satisfied directly from on-hand inventory. Fill rate is one of the most frequently used service measures in practice. You can set a minimum fill rate as a service constraint.
- *Ready Rate* — the probability of no stockout in a review period. You can set a minimum ready rate as a service constraint.
- *Backorder Ratio* — average backorders divided by average demand. You can set a maximum backorder ratio as a service constraint.

These service constraints provide different ways of penalizing backorders. When using fill rate as a service measure, the focus is only on the size of backorders, whereas with backorder ratio as a service measure, the focus is both on the amount and length of backorders. When using ready rate as a service measure, the focus is not on the size and length of backorders, but whether or not a stockout occurs.

Note that setting a high target service level may result in high inventory levels, which can be very costly if demand is intermittent (slow-moving). In these cases, estimating penalty costs and performing a cost optimization may be preferred.

PROC IRP reports several other measures to evaluate the performance of a policy:

- *Average Ordering Frequency* — number of replenishment orders placed per review period. You can set a limit on the average ordering frequency.
- *Average Inventory* — average on-hand inventory at the end of a review period.
- *Average Backorder* — average amount of outstanding backordered demand in a review period.
- *Inventory Ratio* — average inventory divided by average demand.
- *Turnover* — average demand divided by average inventory.
- *Average Cost* — average cost (holding and replenishment) incurred per period. If backorder penalty costs are present, these are included as well.

Lost Sales

A *lost-sales* inventory system allows for unsatisfied demand to be lost rather than backordered. For an (s, S) policy, this system can be approximated by using the fill rate service measure with some slight modifications (Tijms and Groenevelt 1984).

Let β_l represent the fraction of satisfied demand in the lost-sales case. Therefore, $1 - \beta_l$ represents the fraction of demand that is lost. The reorder and order-up-to levels for the lost-sales inventory system are approximately the same as those in a backordering inventory system that has a target fill rate service level specified as $\beta_f = 2 - 1/\beta_l$. This approximation should be used only when β_l is close to 1.

Two-Echelon Distribution System

PROC IRP can find nearly optimal policies for two-echelon distribution systems with different service constraints on multiple retail locations. A two-echelon distribution system consists of a single warehouse and N retail locations. The retail locations pull items from the warehouse and the items are supplied to the warehouse by an exogenous supplier. Figure 2.5 shows a two-echelon distribution system, where node 0 designates the warehouse and nodes 1 through N designate the retail locations.

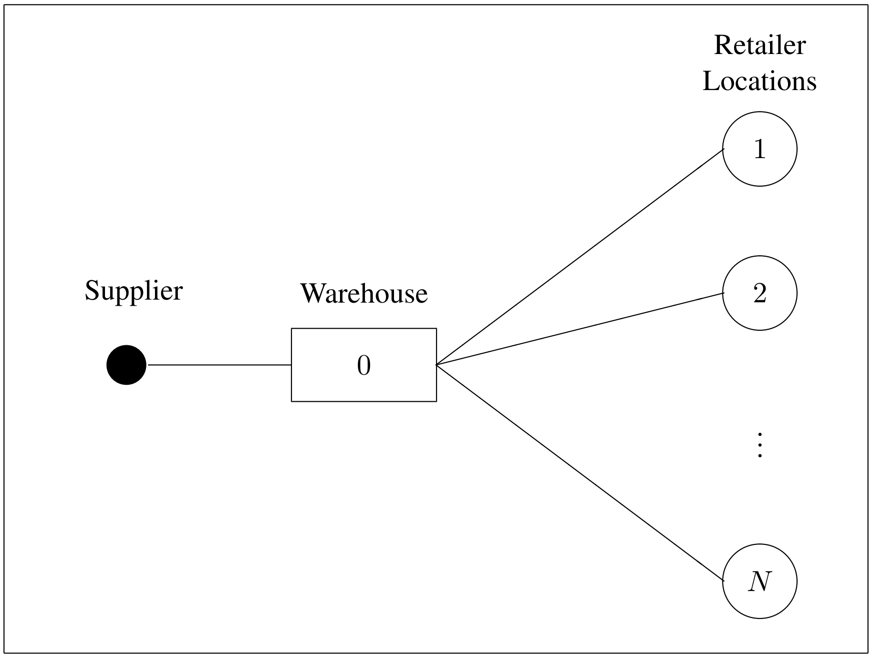


Figure 2.5. Two-Echelon Distribution System

Retail locations place replenishment orders at the warehouse according to a base-stock policy. The replenishment cost of retail locations is negligible or constant. There is a lead time L_i from the warehouse to retailer i. In addition, there is a lead time L_0 from the outside supplier to the warehouse. If the warehouse has sufficient inventory on hand, it immediately dispatches the order, so that the order will arrive at the retail location after the appropriate lead time. If the warehouse has some inventory on hand, but not enough to fill the entire order, it partially fills the order with the on-hand inventory and backorders the rest. If the warehouse has no inventory on hand when a retail location places an order, the entire order is backordered at the warehouse. Note that the average lead time realized at retail location i is greater than L_i since the retail locations have to wait longer if the warehouse is out of stock. All orders received at the warehouse have the same priority. The warehouse follows an (s, S) or (s, nQ) policy and incurs a fixed replenishment cost every time it places an order from the outside supplier. The retail locations can have different demand patterns and service constraints. If the penalty costs on backorders at the retail locations are known, the total system cost incurred per period is minimized.

Policy Algorithm

Single-Location Systems

When the IRP procedure is used to calculate replenishment policies for single-location inventory systems, the underlying assumptions of the optimization model are as follows:

- The holding and stockout costs are linear.
- The probability that replenishment orders cross in time or arrive simultaneously is negligible.

- The stock on hand just after arrival of a replenishment order is positive except for a negligible probability.
- The replenishment lead time is bounded and discrete-valued (integer multiple of the review period).
- Review period demand is independent and identically distributed (stationary). If demand is nonstationary, PROC IRP can still find nearly optimal myopic policies using the LEADTIMEDEMAND statement. See Example 2.4 for an illustration.

Furthermore, if the OPTIMAL option is not used, it is assumed that the review-time demand is not deterministic or nearly deterministic; that is, the coefficient of variation of the review-time demand is not extremely small.

Let

$$
\begin{aligned}
D_R &= \text{demand during review time} \\
D_L &= \text{demand during lead time} \\
L &= \text{lead time (in number of review periods)} \\
D_{LR} &= \text{demand during lead time plus review time} \\
OF &= \text{ordering frequency per period} \\
I &= \text{on-hand inventory at the end of a period} \\
B &= \text{outstanding backorders in a period} \\
K &= \text{fixed cost of replenishment} \\
h &= \text{holding cost per period} \\
p &= \text{penalty cost per period}
\end{aligned}
$$

PROC IRP supports two different methods for solving single-location inventory problems. When the METHOD= option is specified as SERVICE, PROC IRP uses a service level requirement to constrain the optimization. Alternatively, when the METHOD= option is specified as PENALTY, PROC IRP uses backorder penalty costs to drive the optimization.

By default, PROC IRP uses a heuristic algorithm to calculate nearly optimal policies. If the penalty cost method is used, the OPTIMAL option can be specified on the PENALTY statement to indicate that PROC IRP should calculate optimal policies.

The type of policy calculated by the IRP procedure is determined by the value of the POLICYTYPE variable. See the "POLICYTYPE Statement" section on page 25 for more information.

Service Constraint Method

If the METHOD= option is specified as SERVICE, PROC IRP finds nearly optimal policies where the replenishment and holding costs are minimized subject to a service level constraint. The policy calculation is done in three steps.

Step 1: The mean and variance of D_L and D_{LR} are calculated (unless they are specified with the LEADTIMEDEMAND statement).

Step 2: The algorithm finds $\Delta = S - s$ (the gap between S and s for (s, S) policies) if the value of the POLICYTYPE variable is 'SS,' or $\Delta = Q$ (the base lot size for (s, nQ) policies) if the value of the POLICYTYPE variable is 'RQ' or 'NQ.'

If the fixed replenishment cost, K, is known (specified by the FCOST= variable), Δ is determined according to the specification of the DELTA= option. If the DELTA= option is specified as EOQ, Δ is set to the *classic economic order quantity* (EOQ). If the DELTA= option is specified as POWER, a power approximation (similar to the one in Ehrhardt and Mosier 1984) is used to determine Δ.

If the fixed replenishment cost, K, is not known and/or there is a constraint on Δ (specified by the MINSIZE= variable) or a constraint on the ordering frequency (specified by the MAXFREQ= variable), Δ is adjusted so that these constraints are met. If the value of the POLICYTYPE variable is 'RQ' or 'NQ' and a base lot size, Q, is specified (by the LOTSIZE= variable), Δ is set to Q, all other constraints are ignored, and an (s, nQ) policy is calculated. If the value of the POLICYTYPE variable is 'SS' and a base lot size, Q, is specified, then an (s, nQ, S) policy is calculated.

Step 3: The reorder level, s, is found such that the user-specified service type and desired service level are met.

This approach works well for large Δ ($\Delta \geq 1.5 \times E(D_R)$) and leads to nearly optimal solutions. If Δ is small, a modified policy is calculated. See the "Modified Policies" section on page 39 for details.

A suitable distribution must be chosen to represent both lead-time demand and (lead time + review time)-demand distributions. In practice, the normal distribution is widely used to approximate these distributions. However, this choice can lead to very poor results if the coefficients of variation are not small. To overcome this drawback of the normal distribution, PROC IRP uses the gamma distribution if the coefficient of variation of (lead time + review time)-demand is greater than 0.5. In both cases, a two-moment approximation is used.

Penalty Cost Method

If the METHOD= option is specified as PENALTY, PROC IRP finds nearly optimal policies where the replenishment, holding, and backorder penalty costs are minimized. The policy calculation is the same as outlined in the "Service Constraint Method" section on page 37, except for the final step.

In the final step, the reorder level, s, is found such that the average cost per period

$$C(s, \Delta) = K\,E(OF) + h\,E(I) + p\,E(B)$$

is minimized. The choice of distribution used to represent both lead-time demand and (lead time + review time)-demand is the same as described in the "Service Constraint Method" section on page 37.

Note that this heuristic finds Δ and s sequentially. You can specify the OPTIMAL option to find true optimal policies for single-location systems where Δ and s are jointly optimized. See the "OPTIMAL Option" section on page 39 for detailed information.

Modified Policies

If Δ is small ($\Delta < 1.5 \times E(D_R)$), a modified policy will be calculated. For (s, S) policies, the reorder level, s, and order-up-to level, S, are determined as

$$s = \begin{cases} S_b, & S_b < s_p \\ s_p, & s_p \leq S_b \leq S_p \\ S_b - \Delta, & S_b > S_p \end{cases}$$

$$S = S_b$$

where (s_p, S_p) is the policy found assuming Δ is large and S_b is the base stock level for the same problem. If there are constraints on the order size or the ordering frequency, these are taken into account as well. For (s, nQ) policies, the reorder level, s, and base lot size, Q, are determined as

$$s = S_b - \Delta/2$$

$$Q = \Delta$$

where S_b is the base stock level for the same problem. The performance measure estimates reported are those of the base stock policy, with the exception of the average ordering frequency.

This modification is identified by the value MS, MR, or MN in the YY field of the _ALGORITHM_ variable. For details on the _ALGORITHM_ variable, see the "Definitions of Variables in the OUT= Data Set" section on page 30.

Base-Stock Policies

If the value of the POLICYTYPE variable is 'BS,' or if there is no cost of ordering ($K = 0$) and no constraints on the order size or the ordering frequency, a base-stock policy is calculated. The policy calculation is similar to what is outlined in the "Service Constraint Method" section on page 37. The difference is that step 2 is skipped since $\Delta = 0$ for base-stock policies.

OPTIMAL Option

If the OPTIMAL option on the PENALTY statement is specified, PROC IRP finds optimal (s, S) or (s, nQ) policies for single-location systems (see Zheng and Federgruen (1992) and Zheng and Chen (1992) for details). The decision variables are s and S for (s, S) policies and s and Q for (s, nQ) policies. In this case, the variables specified by the LOTSIZE=, MAXFREQ=, and MINSIZE= options on the REPLENISHMENT statement are ignored. The algorithm used when the OPTIMAL option is specified is slower than the heuristic algorithm. Note that the OPTIMAL option is not available for two-echelon distribution systems.

Define the following notation:

$$
\begin{aligned}
C_{SS}(s,S) &= \text{average cost of an } (s,S) \text{ policy} \\
&\quad \text{(when the value of the POLICYTYPE variable is `SS')} \\
C_{NQ}(s,Q) &= \text{average cost of an } (s,nQ) \text{ policy when the fixed cost } K \\
&\quad \text{is incurred for each lot } Q \text{ ordered} \\
&\quad \text{(when the value of the POLICYTYPE variable is `NQ')} \\
C_{RQ}(s,Q) &= \text{average cost of an } (s,nQ) \text{ policy when the fixed cost } K \\
&\quad \text{is incurred independent of the number of lots ordered} \\
&\quad \text{(when the value of the POLICYTYPE variable is `RQ')}
\end{aligned}
$$

In each instance, PROC IRP finds optimal values of the decision variables s^*, S^*, and Q^* such that the average cost per period is minimized:

$$
\begin{aligned}
C_{SS}(s^*,S^*) &= \min_{\forall s,S} C_{SS}(s,S) \\
C_{NQ}(s^*,Q^*) &= \min_{\forall s,Q} C_{NQ}(s,Q) \\
C_{RQ}(s^*,Q^*) &= \min_{\forall s,Q} C_{RQ}(s,Q)
\end{aligned}
$$

Each optimal policy is optimal within its own class. Note that (s,S) policies are optimal among *all* classes of policies for single-location systems:

$$C_{SS}(s^*,S^*) \leq C_{RQ}(s^*,Q^*) \leq C_{NQ}(s^*,Q^*)$$

Suitable distributions must be chosen to represent lead-time demand and review-time demand. These distributions are assumed to be discrete. If the variance is greater than the mean, the distribution under consideration is approximated by a negative binomial distribution. If the variance is less than or equal to the mean, a shifted Poisson distribution is used. The negative binomial and shifted Poisson distributions are fit such that the resulting mean and variance match the mean and variance of the original distribution. The chosen distributions are indicated by a 'B' or 'P' in the 'ZZ' part of the _ALGORITHM_ variable, where the first Z indicates the approximation used for lead-time demand plus review-time demand distribution, and the second Z indicates the approximation used for the review-time demand distribution. While choosing an appropriate distribution, the algorithm may choose a deterministic distribution (a fixed number) to represent these distributions if the variance is close to zero or considerably smaller than the mean. In that case, this number matches the mean of the estimated distribution and this choice is indicated by a 'D' (for deterministic) in the _ALGORITHM_ variable. If the chosen policy is NQ, the review-time demand distribution does not play a role in the optimization algorithm. This is indicated by a '_' in the _ALGORITHM_ variable.

The OUT= data set contains a new variable named _SCALE_, which gives the value used to scale the demand and cost parameters. Initial scaling takes place if the calculated mean of (lead time + review time)-demand is greater than the value specified by the SCALE= option. Further scaling may be performed by the procedure to obtain a suitable fit to the shifted Poisson or negative binomial distribution. If the procedure is unable to find a suitable fit, it stops processing the current item and writes the value "BAD_DATA" to the _STATUS_ variable. Increasing the value of the SCALE= option may resolve this issue.

The magnitude of demand and cost parameters affect the algorithm's memory requirement to calculate policies. In some cases, if insufficient scaling is performed, PROC IRP may run out of memory. In this case PROC IRP stops processing the current item and writes the value "INSUF_MEM" to the _STATUS_ variable. Usually, decreasing the value of the SCALE= option will correct this problem. Note that a smaller value for the SCALE= option will result in scaling by an equal or larger value.

Two-Echelon Distribution Systems

When the IRP procedure is used to calculate replenishment policies for two-echelon distribution systems, the underlying assumptions of the optimization model are the same as in single-location systems. Let

OF_0	$=$	ordering frequency per period at the warehouse
S_0	$=$	order-up-to level at the warehouse
s_i	$=$	reorder level at location $i = 0, \ldots, N$
μ_i	$=$	review-time demand at location $i = 0, \ldots, N$
K_i	$=$	fixed cost of replenishment at location $i = 0, \ldots, N$
h_i	$=$	holding cost per period at location $i = 0, \ldots, N$
p_i	$=$	penalty cost per period at location $i = 1, \ldots, N$
I_i	$=$	on-hand inventory at end of period at location $i = 0, \ldots, N$
B_i	$=$	outstanding backorders in a period at location $i = 0, \ldots, N$

Location $i = 0$ refers to the warehouse.

PROC IRP supports two different methods for solving two-echelon distribution problems. When the METHOD= option is specified as SERVICE, PROC IRP uses a service level requirement to constrain the optimization. Alternatively, when the METHOD= option is specified as PENALTY, PROC IRP uses backorder penalty costs to drive the optimization.

For two-echelon distribution systems, PROC IRP calculates base-stock policies for each retail location. The type of policy for the warehouse is determined by the value of the POLICYTYPE variable.

Service Constraint Method

If the METHOD= option is specified as SERVICE, PROC IRP finds nearly optimal policies where the replenishment and holding costs are minimized subject to service level constraints on the retail locations. The policy calculation is done in three steps.

Step 1 The mean and variance of review-time demand at the warehouse are calculated as the sum of the means and the sum of the variances of review-time demand at the retail locations, respectively. However, if the mean and variance of review-time demand at the warehouse are explicitly specified in the DATA= input data set, those values are used instead.

Note that a collaborative forecast (for all retail locations) may yield a better prediction of the variance of review-time demand at the warehouse than the sum of the variances at the retail locations. Also, note that specifying a value for review-time demand at the warehouse that is significantly different than the sum of the means of review-time demand at the retail locations may cause numerical problems.

Next, the mean and variance of lead-time demand at the warehouse are calculated.

Step 2 This step is the same as for the single-location problem in the "Service Constraint Method" section on page 37. In this step, $\Delta_0 = S_0 - s_0$ is calculated for the warehouse. If Δ_0 is not large ($\Delta_0 < 1.5\mu_0$) and the policy type is specified as 'SS', the following modification is made:

- If $\Delta_0/\mu_0 \leq 0.75$, a base-stock policy at the warehouse is assumed.
- If $0.75 < \Delta_0/\mu_0 < 1.5$, then Δ_0 is set to $1.5\mu_0$, and an (s, S) policy is used at the warehouse.

If a base-stock policy is assumed and if there are constraints on the order size, the policy is modified such that these constraints are satisfied. This modification is identified by the value MS in the _ALGORITHM_ variable.

Step 3 The average cost per period incurred by the system is given as

$$C(s_0, \Delta_0, s_1, s_2, \ldots, s_N) = K_0 E(OF_0) + \sum_{i=0}^{N} h_i E(I_i)$$

In this final step, the cost function is minimized subject to the service level constraint at each retail location. The decision variables are $s_i, i = 0, 1, \ldots, N$. Note that $\Delta_0 = S_0 - s_0$ is calculated and fixed in Step 2.

Each retail location may have a constraint on fill rate, ready rate, or backorder ratio.

As with the single-location problem, a distribution needs to be chosen to represent the lead-time demand and (lead time + review time)-demand at the warehouse and the retail locations. PROC IRP uses the gamma distribution to represent these demand distributions (see the "Service Constraint Method" section on page 37).

Note that PROC IRP ignores any information regarding service levels, service types, and penalty costs at the warehouse, since the backorders at the warehouse are treated implicitly. Similarly, any information regarding policy type, fixed ordering costs, fixed lot size, minimum order size, and maximum ordering frequency are ignored at the retailers since they follow a base-stock policy.

Penalty Cost Method

If the METHOD= option is specified as PENALTY, PROC IRP finds nearly optimal policies where the replenishment, holding, and backorder penalty costs are minimized. The policy calculation is the same as outlined in the "Service Constraint Method" section on page 41 except for the final step.

In the final step, the cost function

$$C(s_0, \Delta_0, s_1, s_2, \ldots, s_N) = K_0 E(OF_0) + \sum_{i=0}^{N} h_i E(I_i) + \sum_{i=1}^{N} p_i E(B_i)$$

is minimized. The decision variables are again $s_i,\ i = 0, 1, \ldots, N$.

Note that this function does not penalize backorders at the warehouse directly (there is no $p_0 E(B_0)$ component). This is justified because customer transactions occur only at the retail locations. Backorders at the warehouse translate to poor performance at the retail locations by increasing the replenishment order lead time.

Retail Location Replenishment Order Lead Time

The replenishment order lead time from the warehouse to any retail location is equal to the shipping and handling time as long as the warehouse has the necessary quantity in stock. In the event of shortages at the warehouse, the retail location has to wait for an extra amount of time, which is dependent upon the time required to replenish the warehouse from a supplier. This extra *warehouse wait* is a function of the warehouse reorder and order-up-to levels. This makes the lead-time demand process at a retail location dependent upon both the retailer base-stock level (s_i) and the warehouse reorder and order-up-to levels (s_0 and S_0). PROC IRP estimates the mean and variance of the warehouse wait using techniques similar to those described in Matta and Sinha (1995).

Examples

This section illustrates how PROC IRP can be used to calculate inventory replenishment policies. Example 2.1 through Example 2.5 focus on a single location system, while Example 2.6 and Example 2.7 focus on a two-echelon distribution system.

Example 2.1. Single Location System: Service Level Heuristic

In this example, inventory replenishment policies are calculated for a single location system using service level heuristics. The retailer purchases finished goods from its suppliers and sells them to its customers. There are ten items to be considered, identified by the SKU variable. Estimates of the holding and fixed costs, mean and variance of the lead time, and mean and variance of the review-time demand are given in Table 2.18. The missing value for Fixed Cost in the last observation indicates that the fixed cost for S10 is difficult to estimate; a maximum order frequency will be placed on this item to account for this.

Table 2.18. Data Estimates for Single Location System

		Holding	Fixed	Lead Time		Review-Time Demand	
SKU	Supplier	Cost	Cost	Mean	Variance	Mean	Variance
S01	ABC Company	0.78	70	1	0.6	39	557
S02	JKL Company	0.96	3	2	1.9	35	404
S03	XYZ Company	0.94	52	2	0	26	199
S04	XYZ Company	0.74	17	3	2.2	75	2541
S05	QRS Company	0.48	19	5	0	9	75
S06	QRS Company	0.68	0	5	6.1	92	4132
S07	ABC Company	0.95	60	2	1.5	94	3266
S08	JKL Company	0.39	90	3	0	20	289
S09	ABC Company	0.47	25	1	0	5	6
S10	ABC Company	0.53	.	4	1.6	62	1437

Based on contracts with its suppliers, the retailer must follow an (s, nQ) policy for items S02, S05, and S08. For items S02 and S08, there is a fixed ordering cost for each lot ordered, and for item S05, there is a single fixed ordering cost, independent of the number of lots ordered. In addition, item S06 has a fixed cost of 0, so the retailer follows a base-stock policy. For the remaining items, the retailer follows (s, S) policies.

The retailer faces additional constraints on some items. When placing an order for item S01, the supplier will not fill any orders that are smaller than 15 items. The supplier for item S07 will only fill orders in multiples of 10 items. The fixed cost of item S10 is unknown, so the retailer imposes a maximum order frequency of 25% (that is, on average, the retailer will order at most once every four review periods). The retailer also imposes a maximum order frequency of 50% for S04, even though the fixed ordering cost is known for this item.

Using this information, the retailer first wants to calculate inventory policies using a target fill rate of 97%. This means that 97% of all incoming customer orders can be filled from on-hand inventory. The information is stored in the data set in1_fr, shown below.

```
data in1_fr;
   format sku $3. supplier $11. policyType $2. serviceType $2.;
   input  sku $ supplier & holdingCost fixedCost LTmean LTvar
          RTDmean RTDvar serviceLevel serviceType $
          policyType $ fixedLotSize minOrderSize maxFreq ;
   datalines;
S01 ABC Company  0.78  70  1  0.6  39   557  0.97  FR  SS   .  15     .
S02 JKL Company  0.96   3  2  1.9  35   404  0.97  FR  NQ   .   .     .
S03 XYZ Company  0.94  52  2  0    26   199  0.97  FR  SS   .   .     .
S04 XYZ Company  0.74  17  3  2.2  75  2541  0.97  FR  SS   .   .  0.50
S05 QRS Company  0.48  19  5  0     9    75  0.97  FR  RQ   .   .     .
S06 QRS Company  0.68   0  5  6.1  92  4132  0.97  FR  BS   .   .     .
S07 ABC Company  0.95  60  2  1.5  94  3266  0.97  FR  SS  10   .     .
S08 JKL Company  0.39  90  3  0    20   289  0.97  FR  NQ   .   .     .
S09 ABC Company  0.47  25  1  0     5     6  0.97  FR  SS   .   .     .
S10 ABC Company  0.53   .  4  1.6  62  1437  0.97  FR  SS   .   .  0.25
;
```

The retailer then uses the following call to PROC IRP to compute the inventory policies. Because the METHOD= option is specified as SERVICE, heuristics are used to compute inventory policies based on target service levels. The variables in the input data set are specified using the HOLDINGCOST, ITEMID, LEADTIME, POLICYTYPE, REPLENISHMENT, REVIEWTIMEDEMAND, and SERVICE statements. Note that two variables, sku and supplier, are specified on the ITEMID statement. The specified variables are copied from the input data set to the output data set; this allows the retailer to include additional information about the suppliers in the output data set.

```
proc irp data=in1_fr out=out1_fr method=service;
   holdingcost holdingCost;
   itemid sku supplier;
   leadtime / mean=LTmean variance=LTvar;
   policytype policyType;
   replenishment / fcost=fixedCost lotsize=fixedLotSize
                   minsize=minOrderSize maxfreq=maxFreq;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel type=serviceType;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.1.1. This data set contains two variables that define the computed policy: reorderLevel and orderUpToLevel. The remaining variables give more details about the policy, including statistics regarding average inventory, average backorders, and so on, as well as the type of algorithm used to compute the policy.

Note that the first two characters in the _algorithm_ variable are 'FR' for all observations; this is because the algorithm used the fill rate target level in the heuristic. The second set of characters in the _algorithm_ variable gives the type of policy computed. For most observations, this is the same as the policyType variable specified in the input data set. However, the policy for item S02 is a modified (s, nQ) policy, and the policy for items S04 and S07 is a modified (s, S) policy; see the "Modified Policies" section on page 39 for more details.

The third set of characters in the _algorithm_ variable indicates which distribution is used to approximate both the lead-time demand and (lead time + review time)-demand. This is either 'GA' for the gamma distribution, or 'NO' for the normal distribution.

Output 2.1.1. Inventory Policies Using 97% Target Fill Rate

PROC IRP Results
Target Measure: 97% Fill Rate

Obs	sku	supplier	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost
1	S01	ABC Company	121	211	99.906	1.70104	0.33507	101.381
2	S02	JKL Company	203	238	117.126	2.12625	1.00000	115.441
3	S03	XYZ Company	91	147	48.283	0.80839	0.35693	63.946
4	S04	XYZ Company	562	663	311.191	4.19117	0.50251	238.824
5	S05	QRS Company	77	118	43.841	0.34130	0.22049	25.233
6	S06	QRS Company	1117	1118	573.694	7.69422	0.99995	390.112
7	S07	ABC Company	513	582	304.933	4.93332	0.76876	335.812
8	S08	JKL Company	103	227	85.694	0.69363	0.16085	47.897
9	S09	ABC Company	9	32	11.780	0.18293	0.19665	10.453
10	S10	ABC Company	418	624	230.778	2.33422	0.25000	122.312

Obs	inventory Ratio	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	2.56168	0.043616	0.39037	0.97007	0.95081	FR-SS-GA	SUCCESSFUL
2	3.34646	0.060750	0.29882	0.97053	0.95210	FR-MN-GA	SUCCESSFUL
3	1.85703	0.031092	0.53849	0.96960	0.93691	FR-SS-NO	SUCCESSFUL
4	4.14922	0.055882	0.24101	0.97088	0.96011	FR-MS-GA	SUCCESSFUL
5	4.87126	0.037922	0.20529	0.96961	0.96356	FR-RQ-NO	SUCCESSFUL
6	6.23581	0.083633	0.16036	0.96998	0.96023	FR-BS-GA	SUCCESSFUL
7	3.24397	0.052482	0.30826	0.97086	0.95530	FR-MS-GA	SUCCESSFUL
8	4.28468	0.034682	0.23339	0.96983	0.95924	FR-NQ-NO	SUCCESSFUL
9	2.35600	0.036586	0.42445	0.96379	0.92373	FR-SS-NO	SUCCESSFUL
10	3.72222	0.037649	0.26866	0.96995	0.95618	FR-SS-NO	SUCCESSFUL

The fill rates for all items are near 97%, the specified target level. However, suppose the retailer thinks the resulting backorder ratios are unacceptably high. Only one service measure per observation can be specified in a single call to PROC IRP, so now the retailer will specify a 3% target backorder ratio for all items, which ignores the 97% target fill rate. The data step used to make this change is as follows.

```
data in1_br;
   set in1_fr;
   serviceLevel = 0.03;
   serviceType = 'BR';
run;
```

The retailer then calls PROC IRP as follows. Note that this call is exactly the same as the previous call to PROC IRP, except for a different name for the output data set. Some of the variable values (for the serviceLevel and serviceType variables) have changed, but the variable names have not changed.

```
proc irp data=in1_br out=out1_br method=service;
   holdingcost holdingCost;
   itemid sku supplier;
   leadtime / mean=LTmean variance=LTvar;
   policytype policyType;
   replenishment / fcost=fixedCost lotsize=fixedLotSize
                   minsize=minOrderSize  maxfreq=maxFreq;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel type=serviceType;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.1.2. Notice that the average inventory increased for the 3% backorder ratio target level, as compared to the 97% fill rate target level. More inventory is required to meet this more restrictive target service measure.

Output 2.1.2. Inventory Policies Using 3% Target Backorder Ratio

PROC IRP Results
Target Measure: 3% Backorder Ratio

Obs	sku	supplier	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost
1	S01	ABC Company	134	224	112.368	1.16393	0.33507	111.102
2	S02	JKL Company	234	268	147.047	1.04651	1.02941	144.253
3	S03	XYZ Company	91	147	48.283	0.80839	0.35693	63.946
4	S04	XYZ Company	630	731	373.259	2.25886	0.50251	284.754
5	S05	QRS Company	79	120	45.775	0.27482	0.22049	26.161
6	S06	QRS Company	1311	1312	762.757	2.75746	0.99995	518.675
7	S07	ABC Company	580	643	363.819	2.81858	0.79908	393.573
8	S08	JKL Company	105	230	88.111	0.61110	0.16085	48.839
9	S09	ABC Company	9	32	11.780	0.18293	0.19665	10.453
10	S10	ABC Company	430	635	242.129	1.85659	0.25000	128.328

Obs	inventory Ratio	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	2.88124	0.029844	0.34707	0.97987	0.96572	BR-SS-GA	SUCCESSFUL
2	4.20133	0.029900	0.23802	0.98663	0.97573	BR-MN-GA	SUCCESSFUL
3	1.85703	0.031092	0.53849	0.96960	0.93691	BR-SS-NO	SUCCESSFUL
4	4.97678	0.030118	0.20093	0.98455	0.97788	BR-MS-GA	SUCCESSFUL
5	5.08609	0.030536	0.19661	0.97515	0.96980	BR-RQ-NO	SUCCESSFUL
6	8.29084	0.029972	0.12062	0.98986	0.98510	BR-BS-GA	SUCCESSFUL
7	3.87041	0.029985	0.25837	0.98388	0.97383	BR-MS-GA	SUCCESSFUL
8	4.40556	0.030555	0.22699	0.97300	0.96341	BR-NQ-NO	SUCCESSFUL
9	2.35600	0.036586	0.42445	0.96379	0.92373	BR-SS-NO	SUCCESSFUL
10	3.90530	0.029945	0.25606	0.97573	0.96401	BR-SS-NO	SUCCESSFUL

As the average inventory increases, so does the average cost. This is because the retailer has not specified a penalty cost for the backorders in order to balance the holding cost of inventory. The only costs considered in the service level heuristics are fixed ordering costs and inventory holding costs. You can see from Output 2.1.1 and Output 2.1.2 that the average ordering frequency (avgOrderFreq) does not change much between the two target level specifications. Therefore, the bulk of the increase in average cost comes from the increase in average inventory.

The retailer now has two policies to choose from. While one policy does have a higher average cost, the decision should not be based on cost alone. Both policies are heuristic policies, and are derived using different target service levels. With a higher level of service comes a higher cost, and the retailer must decide based on the desired levels of service which policy best fits the needs of the company.

Another option for the retailer is to use backorder penalty cost information to find inventory policies. This problem is explored in Example 2.2 and Example 2.3.

Example 2.2. Single Location System: Penalty Costs

In this example, assume that the retailer from Example 2.1 is able to obtain estimates of backorder penalty costs. Rather than using a service level heuristic, as in Example 2.1, the retailer will use the penalty costs to calculate inventory policies. First, the retailer uses a heuristic method to calculate nearly optimal inventory policies. In Example 2.3, the optimal inventory policy will be calculated.

The backorder penalty costs are given in the following data set.

```
data pcosts;
   format sku $3. penaltyCost;
   input sku $ penaltyCost;
   datalines;
S01   7.4
S02  10.2
S03   8.1
S04   6.6
S05   9.2
S06   9.0
S07   7.1
S08   3.7
S09   5.2
S10  10.8
;
```

This data set is merged with in1_fr to produce the input data set in2. The serviceType and serviceLevel variables are dropped from the in1_fr data set, as these will not be needed when using penalty costs. However, if these variables are left in the data set, they will simply be ignored when the METHOD= option is specified as PENALTY.

```
data in2;
   merge in1_fr (drop=serviceType serviceLevel)
         pcosts;
   by sku;
run;
```

The retailer then calls PROC IRP using the following code. There are several differences between this call and the calls to PROC IRP in Example 2.1. First, the METHOD= option is specified as PENALTY. Second, the PENALTY statement is included, and the penalty cost variable is identified as penaltyCost. Note that there are no other options specified in the PENALTY statement, so the policy is calculated using a heuristic. Finally, the SERVICE statement is no longer listed in the PROC IRP call; if it had been listed, it would be ignored.

```
proc irp data=in2 out=out2 method=penalty;
   holdingcost holdingCost;
   itemid sku supplier;
   leadtime / mean=LTmean variance=LTvar;
   penalty / cost=penaltyCost;
   policytype policyType;
   replenishment / fcost=fixedCost lotsize=fixedLotSize
                   minsize=minOrderSize maxfreq=maxFreq;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.2.1. For all items in this example, the average inventory is lower than that shown in Output 2.1.1 and Output 2.1.2, and the average backorders are higher. As a result, the fill rates for this policy are less than 97%, and the backorder ratios are greater than 3%.

The average cost for the penalty cost policy may be higher than that of the previous policies (as is the case for items S03, S05, S09, and S10), lower than that of the previous policies (as is the case for item S07), or may lie between the average costs for the two previous policies (as is the case for the remaining items). For example, the average cost may be lower because PROC IRP was able to find a better policy using the penalty cost heuristic. If the penalty costs are not very high, the resulting total cost may actually be lower. On the other hand, the penalty cost heuristics include penalty costs for backorders; these costs are not included in the service level heuristics, so an increase in average cost for the penalty cost method may result from including this extra cost parameter. Therefore, caution should be used when comparing output from the service level method and penalty cost method, as the two methods use different levels of information to compute costs and determine policies.

Output 2.2.1. Inventory Policies Using Penalty Cost Heuristic

PROC IRP Results
Penalty Cost Heuristic

Obs	sku	supplier	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost
1	S01	ABC Company	96	186	76.659	3.4540	0.33507	108.808
2	S02	JKL Company	175	209	90.957	3.9568	1.02941	130.766
3	S03	XYZ Company	83	139	40.935	1.4610	0.35693	68.874
4	S04	XYZ Company	467	568	211.277	11.2771	0.50251	239.317
5	S05	QRS Company	74	114	40.478	0.4782	0.22049	28.019
6	S06	QRS Company	998	999	461.089	14.0894	0.99995	440.345
7	S07	ABC Company	407	465	196.823	13.8235	0.82395	334.566
8	S08	JKL Company	84	208	67.924	1.9243	0.16085	48.087
9	S09	ABC Company	9	31	11.665	0.1829	0.19665	11.350
10	S10	ABC Company	414	619	226.787	2.5152	0.25000	147.362

Obs	inventory Ratio	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	1.96560	0.08856	0.50875	0.93790	0.90432	PC-SS-GA	SUCCESSFUL
2	2.59876	0.11305	0.38480	0.94230	0.91370	PC-MN-GA	SUCCESSFUL
3	1.57444	0.05619	0.63515	0.94590	0.89771	PC-SS-NO	SUCCESSFUL
4	2.81703	0.15036	0.35498	0.92113	0.89897	PC-MS-GA	SUCCESSFUL
5	4.49758	0.05314	0.22234	0.95943	0.95116	PC-RQ-NO	SUCCESSFUL
6	5.01184	0.15315	0.19953	0.94397	0.92963	PC-BS-GA	SUCCESSFUL
7	2.09387	0.14706	0.47759	0.91546	0.88216	PC-MS-GA	SUCCESSFUL
8	3.39622	0.09622	0.29445	0.92362	0.90599	PC-NQ-NO	SUCCESSFUL
9	2.33299	0.03659	0.42863	0.96379	0.92373	PC-SS-NO	SUCCESSFUL
10	3.65786	0.04057	0.27338	0.96779	0.95329	PC-SS-NO	SUCCESSFUL

Example 2.3. Single Location System: OPTIMAL Option

In Example 2.2, the retailer used penalty costs to compute nearly optimal inventory replenishment policies. By specifying the OPTIMAL option on the PENALTY statement, you can compute optimal policies using backorder penalty costs. PROC IRP computes the optimal reorder level and order-up-to-level within the class of policy specified by the policyType variable (that is, SS, BS, NQ, or RQ).

The call to PROC IRP is shown below. The OPTIMAL option is specified on the PENALTY statement. In addition, the LOTSIZE=, MINSIZE=, and MAXFREQ= options are no longer included on the REPLENISHMENT statement, since these options are ignored when the OPTIMAL option is used.

```
proc irp data=in2 out=out3 method=penalty;
   holdingcost holdingCost;
   itemid sku supplier;
   leadtime / mean=LTmean variance=LTvar;
   penalty / cost=penaltyCost optimal;
   policytype policyType;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.3.1. Notice that the average cost for some items (S01, S06, and S07) is lower than the average cost in Output 2.2.1. This may be expected, as the OPTIMAL option finds the optimal (i.e., lowest-cost) inventory replenishment policy. However, the average cost for the remaining items actually rises. There are two reasons why this might happen. First, the penalty cost heuristic given in Output 2.2.1 uses an approximation to the cost of the policy; the actual cost may be slightly higher or lower than the value given in the avgCost variable. Moreover, the heuristic uses either a gamma distribution or a normal distribution to approximate both the lead-time demand and (lead time + review time)-demand, whereas the optimization uses either a negative binomial distribution or a shifted Poisson distribution. Therefore, the underlying assumptions of the models are different, and care should be used when comparing results across the two models. The policy calculated when using the OPTIMAL option is the optimal policy with respect to the lead time and demand distributions used by PROC IRP, but may reflect a higher cost than a policy calculated using different distributions for lead time and demand.

In this example, the negative binomial distribution is used for the (lead time + review time)-demand of all items, as indicated by a 'B' in the fifth character of the _algorithm_ variable. This distribution is also used for the review-time demand, as indicated by a 'B' in the sixth character of the _algorithm_ variable. Note that the sixth character of the _algorithm_ variable is '_' for items following an 'NQ' policy, which indicates that the review-time demand distribution does not play a role in the optimization algorithm.

Recall from Example 2.1 that the fixed cost for item S10 was not easily estimated, so a maximum ordering frequency was used instead. However, the OPTIMAL option ignores the LOTSIZE=, MINSIZE=, and MAXFREQ= options, so item S10 is no longer constrained by a maximum ordering frequency of 25%. In addition, because the fixed cost for item S10 was not specified, PROC IRP assumes it is zero. As a result, the policy for S10 in Output 2.3.1 is a base-stock policy (as indicated by the 'BS' in the _algorithm_ variable), and the reorderLevel and orderUpToLevel are quite different from those in the previous examples. However, the original intention of including a missing value for the fixed cost for S10 was to account for the fact that the cost was unknown, not to imply that the cost was zero. Therefore, when using the OPTIMAL option, you should specify estimates for fixed costs of all items, unless the fixed cost is assumed to be zero.

Output 2.3.1. Inventory Policies Using the OPTIMAL Option

PROC IRP Results
Penalty Cost with OPTIMAL Option

Obs	sku	supplier	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost
1	S01	ABC Company	93	188	77.538	3.4709	0.32194	108.700
2	S02	JKL Company	173	213	92.623	3.8979	0.87719	131.308
3	S03	XYZ Company	83	138	41.674	1.7570	0.36451	72.360
4	S04	XYZ Company	441	525	214.188	11.2445	0.54722	242.016
5	S05	QRS Company	76	117	43.684	0.6844	0.21791	31.405
6	S06	QRS Company	999	1000	461.055	13.9348	0.99252	438.931
7	S07	ABC Company	375	522	206.377	13.8543	0.44856	321.338
8	S08	JKL Company	83	208	68.445	2.4453	0.16000	50.141
9	S09	ABC Company	8	31	11.429	0.2308	0.19531	11.455
10	S10	ABC Company	526	527	220.297	3.2970	0.99800	152.365

Obs	inventory Ratio	backorder Ratio	turnover	ready Rate	_scale_	_algorithm_	_status_
1	1.98815	0.08900	0.50298	0.90281	1.00	PC-SS-BB	SUCCESSFUL
2	2.64637	0.11137	0.37788	0.91369	1.05	PC-NQ-B_	SUCCESSFUL
3	1.60286	0.06758	0.62388	0.89413	1.00	PC-SS-BB	SUCCESSFUL
4	2.85584	0.14993	0.35016	0.89768	3.00	PC-SS-BB	SUCCESSFUL
5	4.85382	0.07605	0.20602	0.94879	1.00	PC-RQ-BB	SUCCESSFUL
6	5.01147	0.15147	0.19954	0.92892	5.52	PC-BS-BB	SUCCESSFUL
7	2.19550	0.14739	0.45548	0.88118	2.82	PC-SS-BB	SUCCESSFUL
8	3.42226	0.12226	0.29220	0.90366	1.00	PC-NQ-B_	SUCCESSFUL
9	2.28584	0.04615	0.43748	0.90048	1.00	PC-SS-BB	SUCCESSFUL
10	3.55318	0.05318	0.28144	0.95311	3.10	PC-BS-BB	SUCCESSFUL

Example 2.4. Single Location System: LEADTIMEDEMAND Statement

This example illustrates the use of PROC IRP for a retailer that faces nonstationary demand with a lead time that is longer than the review period. The IRP procedure uses the review-time demand and lead time information to calculate the parameters of lead-time demand. When demand is nonstationary (i.e., demand fluctuates over time), it is not sufficient to know just the lead time and mean review-time demand information. In such situations, you can directly specify the mean and variance of lead-time demand with the LEADTIMEDEMAND statement.

For example, suppose the lead time for an item is three periods, but the demands over the next four review periods are 25, 32, 40, and 28. If the mean of the review-time demand is specified as 25 (the mean of the current period's demand), and the lead-time mean is specified as 3 using the LEADTIME statement, PROC IRP will compute the mean lead-time demand as 75 (= 3×25). This is an inaccurate calculation of lead-time demand, as it does not account for the fluctuations in demand in the subsequent periods. Rather, the correct calculation of lead-time demand is the demand over the next three periods following the current review period, which is 100 (= 32 + 40 + 28). This example illustrates how the LEADTIMEDEMAND statement is used to overcome such a problem.

The data set in4 gives the input to PROC IRP. The mean and variance of lead-time demand are given by the LTDmean and LTDvar variables, respectively. The mean and variance of review-time demand are given by the RTDmean and RTDvar variables, respectively. Note that the mean and variance of lead time are not included in this data set. When using the LEADTIMEDEMAND statement, these variables are not used.

```
data in4;
   format sku $3.;
   input  sku $ holdingCost fixedCost
          LTDmean LTDvar RTDmean RTDvar
          serviceLevel;
datalines;
B01  0.52  62  100   894  25   56  0.95
B02  0.86  17   80   633  50  227  0.95
B03  0.27  48  275  4101  90  506  0.95
B04  0.94  23   64   719  15   38  0.95
B05  0.62  38   90  1188  32  163  0.95
B06  0.44  82  122  4324  52  675  0.95
B07  0.75  68  170  2823  84  632  0.95
B08  0.78  73   30   365  10   35  0.95
B09  0.46  18   91   989  66  533  0.95
B10  0.55  25  144  3741  71  807  0.95
;
```

The following call to PROC IRP computes inventory replenishment policies using a 95% target fill rate. In the PROC IRP statement, the METHOD= option is not specified, so the default value of SERVICE is used. The HOLDINGCOST statement is not required because the holding cost variable is named holdingCost, the default name for PROC IRP. Because the POLICYTYPE statement is not specified, PROC IRP computes (s, S) policies (or modified (s, S) policies) for all items in the data set. In addition, the REPLENISHMENT statement is not required because the fixed cost variable is named fixedCost, the default name for PROC IRP. Finally, the SERVICE statement is not specified, so PROC IRP uses fill rate, the default service measure, for all items.

```
proc irp data=in4 out=out4;
   itemid sku;
   leadtimedemand / mean=LTDmean variance=LTDvar;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.4.1. Note that because the LEADTIMEDEMAND statement was used, these policies should be interpreted as the policies to follow only for the current review period. Because demand is nonstationary and the lead times are longer than the review period, you should compute new policies each period, using updated information about lead-time demand and review-time demand.

Output 2.4.1. Inventory Policies Using the LEADTIMEDEMAND Statement

PROC IRP Results
Target Measure: 95% Fill Rate

Obs	sku	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost	inventory Ratio
1	B01	134	216	57.386	1.52076	0.26127	46.039	2.29542
2	B02	133	159	31.495	2.49541	0.88356	42.107	0.62991
3	B03	367	550	117.110	4.76695	0.39053	50.365	1.30122
4	B04	102	134	43.416	0.99821	0.36862	49.289	2.89439
5	B05	139	206	59.673	1.76624	0.37202	51.134	1.86477
6	B06	214	366	133.069	3.81690	0.28245	81.711	2.55901
7	B07	259	386	89.724	4.35143	0.48567	100.319	1.06815
8	B08	52	102	40.822	1.03108	0.17768	44.811	4.08216
9	B09	159	196	42.252	3.25241	0.86719	35.046	0.64019
10	B10	250	297	85.662	3.66162	0.82287	67.686	1.20650

Obs	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	0.06083	0.43565	0.94950	0.91308	FR-SS-NO	SUCCESSFUL
2	0.04991	1.58753	0.95021	0.83864	FR-MS-NO	SUCCESSFUL
3	0.05297	0.76851	0.94969	0.88679	FR-SS-NO	SUCCESSFUL
4	0.06655	0.34550	0.95005	0.92375	FR-SS-NO	SUCCESSFUL
5	0.05520	0.53626	0.95082	0.91043	FR-SS-NO	SUCCESSFUL
6	0.07340	0.39078	0.94975	0.92376	FR-SS-GA	SUCCESSFUL
7	0.05180	0.93620	0.94960	0.87838	FR-SS-NO	SUCCESSFUL
8	0.10311	0.24497	0.94958	0.93132	FR-SS-GA	SUCCESSFUL
9	0.04928	1.56204	0.95077	0.84127	FR-MS-NO	SUCCESSFUL
10	0.05157	0.82884	0.95015	0.88799	FR-MS-NO	SUCCESSFUL

Example 2.5. Continuous Review Approximation: Review Period Shorter than Forecast Interval

In this example, consider a retailer that forecasts demand data on a monthly basis but reviews inventory on a weekly basis. For the purpose of this illustration, it is assumed that there are exactly four weeks in a month.

For example, consider Table 2.1 on page 14 in the "Getting Started" section, but suppose that the mean and variance of demand specify the demand over one month. In addition, suppose the lead time of all items is one week (the same as the review period). This is not an example of continuous review, since the retailer still makes decisions at discrete time periods. However, it may be considered an approximation to a continuous review system, because decisions are made at points throughout the demand forecast interval. By choosing smaller review periods (for example, one day or one hour), this becomes a closer approximation to a continuous review system.

The data for this example are given in the following data set, data5. The variables MeanOfDemand and VarianceOfDemand give the mean and variance of demand over an entire month.

```
data data5;
   format Sku $1.;
   input  Sku $ HoldingCost OrderingCost
          LeadTime MeanOfDemand VarianceOfDemand;
datalines;
A  0.35  90  1  125.1  2170.8
B  0.05  50  1  140.3  1667.7
C  0.12  50  1  116.0  3213.4
D  0.10  75  1  291.8  5212.4
E  0.45  75  1  134.5  1980.5
;
```

This data set is transformed to the input data set for PROC IRP using the following data step. From the assumption that there are four weeks in a month, the mean and variance of review-time demand (RTDmean and RTDvar, respectively) are calculated by dividing MeanOfDemand by 4 and VarianceOfDemand by 16. For this calculation to be valid, the demand for one month must be assumed uniform over the entire month, so that the demand for a single week is one quarter of the monthly demand.

```
data in5;
   set data5;
   RTDmean = MeanOfDemand / 4 ;
   RTDvar = VarianceOfDemand / 16 ;
   serviceLevel = 0.96 ;
run;
```

The call to PROC IRP is as follows. Notice that the VARIANCE= option is not specified on the LEADTIME statement, since the lead times are assumed to be deterministic (that is, zero variance).

```
proc irp data=in5 out=out5 method=service;
   holdingcost HoldingCost;
   itemid Sku;
   leadtime / mean=LeadTime;
   replenishment / fcost=OrderingCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel;
run;
```

The output data set from this call to PROC IRP is given in Output 2.5.1.

Output 2.5.1. Inventory Policies When Review Period Is Shorter than Forecast Interval

PROC IRP Results
Target Measure: 96% Fill Rate

Obs	Sku	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq	avgCost	inventory Ratio
1	A	52	173	59.004	1.22044	0.22500	40.9018	1.88663
2	B	47	289	108.019	1.41980	0.13446	12.1238	3.07966
3	C	49	198	74.775	1.16642	0.17345	17.6454	2.57845
4	D	109	417	137.466	2.87227	0.20995	29.4925	1.88439
5	E	57	158	49.747	1.30180	0.27889	43.3032	1.47947

Obs	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	0.039023	0.53005	0.96115	0.90539	FR-SS-NO	SUCCESSFUL
2	0.040479	0.32471	0.95981	0.90998	FR-SS-NO	SUCCESSFUL
3	0.040221	0.38783	0.96037	0.92052	FR-SS-NO	SUCCESSFUL
4	0.039373	0.53068	0.96066	0.89137	FR-SS-NO	SUCCESSFUL
5	0.038715	0.67592	0.96135	0.89476	FR-SS-NO	SUCCESSFUL

Example 2.6. Two-Echelon System: Service Level Heuristic

This example illustrates the use of PROC IRP to compute inventory replenishment policies for a two-echelon system using service level heuristics. Example 2.7 then explores the same two-echelon system using penalty costs.

In this example, there are two warehouses and four retailers. There are two items (M01 and M02), but these items can be classified further by color (BLUE or RED). These items could be identified by four distinct values of the SKU variable; however, to illustrate the use of the ITEMID statement, they will be identified by SKU (M01 or M02) and COLOR (BLUE or RED). Warehouse W01 supplies item M01, and warehouse W02 supplies item M02. From warehouse W01, only retailers R01 and R02 require blue items, while retailers R01, R03, and R04 require red items. From warehouse W02, all four retailers (R01, R02, R03, and R04) require blue items, while only retailer R03 requires red items.

Estimates of the holding and fixed costs, mean and variance of the lead time, and mean and variance of the review-time demand are given in Table 2.19. Observations with a missing value for Retailer correspond to warehouses. For example, the first observation gives the holding cost and fixed cost at warehouse W01, in addition to the mean and variance of lead time from an external supplier to this warehouse. The mean and variance of review-time demand are missing for all the warehouse observations, since the warehouses do not see any external demand apart from the orders placed by the retailers.

The remaining observations correspond to retailers. The missing values of Fixed Cost indicate that the retailers incur no fixed cost for placing an order; therefore, the retailers follow base-stock policies. For these observations, the mean and variance of lead time give data about the lead time from the warehouse to the retailer. In this problem, the lead time variance between the warehouses and the retailers is assumed to be zero, but positive variances can also be used in the two-echelon system. The review-time demand data give the mean and variance of the demand for that item (SKU and Color) at that retailer.

Table 2.19. Data Estimates for Two-Echelon System

SKU	Color	Warehouse	Retailer	Holding Cost	Fixed Cost	Lead Time Mean	Lead Time Variance	Review-Time Demand Mean	Review-Time Demand Variance
M01	Blue	W01	.	0.20	61	1	0.12	.	.
M01	Blue	W01	R01	0.75	.	2	0	67	121
M01	Blue	W01	R02	1.42	.	1	0	23	87
M01	Red	W01	.	0.20	61	1	0.12	.	.
M01	Red	W01	R01	0.75	.	2	0	50	793
M01	Red	W01	R03	1.11	.	3	0	42	109
M01	Red	W01	R04	0.65	.	2	0	91	1267
M02	Blue	W02	.	0.17	88	1	0.41	.	.
M02	Blue	W02	R01	0.70	.	1	0	84	931
M02	Blue	W02	R02	1.35	.	2	0	59	1018
M02	Blue	W02	R03	1.04	.	1	0	71	775
M02	Blue	W02	R04	0.62	.	2	0	113	1689
M02	Red	W02	.	0.17	88	1	0.41	.	.
M02	Red	W02	R03	1.04	.	1	0	85	1954

Using this information, inventory policies are calculated using (s, S) policies for the warehouses and a target fill rate of 97% for the retailers. This means that 97% of all incoming customer orders (to the retailers) can be filled from on-hand inventory. The information is stored in the data set in6_fr, which is shown below.

```
data in6_fr;
   format warehouse $3. retailer $3. sku $3. color $4.
          policyType $2. serviceType $2. ;
   input sku $ color $ warehouse $ retailer $
         holdingCost fixedCost
         LTmean LTvar RTDmean RTDvar
         policyType $ serviceType $ serviceLevel;
   datalines;
M01  BLUE  W01    .  0.20  61  1  0.12    .     .  SS   .     .
M01  BLUE  W01  R01  0.75   .  2  0      67   121   .  FR  0.97
M01  BLUE  W01  R02  1.42   .  1  0      23    87   .  FR  0.97
M01  RED   W01    .  0.20  61  1  0.12    .     .  SS   .     .
M01  RED   W01  R01  0.75   .  2  0      50   793   .  FR  0.97
M01  RED   W01  R03  1.11   .  3  0      42   109   .  FR  0.97
M01  RED   W01  R04  0.65   .  2  0      91  1267   .  FR  0.97
M02  BLUE  W02    .  0.17  88  1  0.41    .     .  SS   .     .
M02  BLUE  W02  R01  0.70   .  1  0      84   931   .  FR  0.97
M02  BLUE  W02  R02  1.35   .  2  0      59  1018   .  FR  0.97
M02  BLUE  W02  R03  1.04   .  1  0      71   775   .  FR  0.97
M02  BLUE  W02  R04  0.62   .  2  0     113  1689   .  FR  0.97
M02  RED   W02    .  0.17  88  1  0.41    .     .  SS   .     .
M02  RED   W02  R03  1.04   .  1  0      85  1954   .  FR  0.97
;
```

The following call to PROC IRP computes the inventory policies. Because the METHOD= option is specified as SERVICE, heuristics are used to compute inventory policies based on target service levels (in this case, 97% fill rate). The ITEMID statement specifies sku, color, and warehouse as the variables by which to group the items. The LOCATION statement specifies the retailer variable. The remaining variables in the input data set are specified using the HOLDINGCOST, LEADTIME, POLICYTYPE, REPLENISHMENT, REVIEWTIMEDEMAND, and SERVICE statements.

```
proc irp data=in6_fr out=out6_fr method=service;
   holdingcost holdingCost;
   itemid sku color warehouse;
   leadtime / mean=LTmean variance=LTvar;
   location retailer;
   policytype policyType;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel type=serviceType;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.6.1. This data set contains two variables that define the computed policy: reorderLevel and orderUpToLevel. The retailers follow base-stock policies, so the orderUpToLevel is one more than the reorderLevel for the retailers. Note that the value of fillRate for the retailers is very near 97%, the target service level. The fill rate for the warehouses is lower, but the fill rate at the retailers is the main concern since customers are only seen at the retailers.

Output 2.6.1. Inventory Policies Using 97% Target Fill Rate

PROC IRP Results for Two-Echelon System
Target Measure: 97% Fill Rate

Obs	sku	color	warehouse	retailer	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq
1	M01	BLUE	W01		143	370	100.495	4.9359	0.32990
2	M01	BLUE	W01	R01	227	228	25.243	1.9699	1.00000
3	M01	BLUE	W01	R02	67	68	21.412	0.6913	1.00000
4	M01	RED	W01		295	627	147.610	14.8469	0.42581
5	M01	RED	W01	R01	243	244	91.488	1.6529	1.00000
6	M01	RED	W01	R03	201	202	31.761	1.2589	1.00000
7	M01	RED	W01	R04	379	380	102.313	2.8924	1.00000
8	M02	BLUE	W02		653	1247	382.574	20.5227	0.42808
9	M02	BLUE	W02	R01	241	242	71.310	2.5229	1.00000
10	M02	BLUE	W02	R02	280	281	102.252	1.9133	1.00000
11	M02	BLUE	W02	R03	210	211	66.704	2.1103	1.00000
12	M02	BLUE	W02	R04	459	460	117.395	3.4087	1.00000
13	M02	RED	W02		191	494	199.636	4.3689	0.23837
14	M02	RED	W02	R03	288	289	117.160	2.6411	1.00000

Obs	avgCost	inventory Ratio	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	89.559	1.11662	0.054843	0.89556	0.94664	0.85528	__-SS-GA	SUCCESSFUL
2	18.932	0.37676	0.029401	2.65424	0.97060	0.84850	FR-BS-GA	SUCCESSFUL
3	30.405	0.93095	0.030055	1.07418	0.97027	0.91984	FR-BS-GA	SUCCESSFUL
4	225.870	0.80661	0.081131	1.23976	0.92210	0.81265	__-SS-GA	SUCCESSFUL
5	68.616	1.82976	0.033057	0.54652	0.96946	0.94678	FR-BS-GA	SUCCESSFUL
6	35.254	0.75620	0.029974	1.32239	0.97025	0.90077	FR-BS-GA	SUCCESSFUL
7	66.503	1.12432	0.031785	0.88943	0.96904	0.92432	FR-BS-GA	SUCCESSFUL
8	432.822	1.16995	0.062761	0.85474	0.94806	0.88077	__-SS-GA	SUCCESSFUL
9	49.917	0.84892	0.030034	1.17796	0.97030	0.91414	FR-BS-GA	SUCCESSFUL
10	138.040	1.73308	0.032429	0.57701	0.96978	0.94516	FR-BS-GA	SUCCESSFUL
11	69.372	0.93949	0.029722	1.06441	0.97071	0.92091	FR-BS-GA	SUCCESSFUL
12	72.785	1.03890	0.030165	0.96256	0.97043	0.92242	FR-BS-GA	SUCCESSFUL
13	176.761	2.34866	0.051398	0.42577	0.95939	0.93192	__-SS-GA	SUCCESSFUL
14	121.846	1.37835	0.031072	0.72550	0.97021	0.93966	FR-BS-GA	SUCCESSFUL

Suppose that the target service measure for retailer R01 is instead specified as a 3% backorder ratio. The remaining retailers continue to follow policies based on a 97% target fill rate. The data step to change the serviceType and serviceLevel variables is as follows.

```
data in6_br;
   set in6_fr;
   if retailer = 'R01' then do;
      serviceType = 'BR';
      serviceLevel = 0.03;
   end;
run;
```

The call to PROC IRP is as follows. Note that this call is exactly the same as the previous call to PROC IRP, except for a different name for the output data set. Some of the variable values (for the serviceLevel and serviceType variables) have changed, but the variable names have not changed.

```
proc irp data=in6_br out=out6_br method=service;
   holdingcost holdingCost;
   itemid sku color warehouse;
   leadtime / mean=LTmean variance=LTvar;
   location retailer;
   policytype policyType;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
   service / level=serviceLevel type=serviceType;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.6.2. The values of the backorderRatio variable are near 3% for the three observations that correspond to retailer R01. Note that when the policies for retailer R01 change, the policies for the other retailers and the policies for the warehouses may also change as a result of the changes in the target service level for R01.

Output 2.6.2. Inventory Policies Using 3% Target Backorder Ratio for R01

```
                    PROC IRP Results for Two-Echelon System
                     Target Measure:  3% Backorder Ratio

                                            order                     avg
                                   reorder  UpTo      avg       avg   Order
Obs sku color warehouse retailer  Level    Level  Inventory Backorder  Freq

  1 M01 BLUE     W01                143      370   100.495    4.9359  0.32990
  2 M01 BLUE     W01      R01       227      228    25.243    1.9699  1.00000
  3 M01 BLUE     W01      R02        67       68    21.412    0.6913  1.00000
  4 M01 RED      W01                286      618   140.370   16.6069  0.42581
  5 M01 RED      W01      R01       247      248    95.024    1.4843  1.00000
  6 M01 RED      W01      R03       202      203    32.471    1.2179  1.00000
  7 M01 RED      W01      R04       382      383   104.645    2.7621  1.00000
  8 M02 BLUE     W02                664     1258   392.298   19.2468  0.42808
  9 M02 BLUE     W02      R01       240      241    70.570    2.5478  1.00000
 10 M02 BLUE     W02      R02       280      281   102.394    1.8901  1.00000
 11 M02 BLUE     W02      R03       209      210    65.935    2.1425  1.00000
 12 M02 BLUE     W02      R04       457      458   115.803    3.4986  1.00000
 13 M02 RED      W02                191      494   199.636    4.3689  0.23837
 14 M02 RED      W02      R03       288      289   117.160    2.6411  1.00000


            inventory backorder            fill    ready
Obs avgCost   Ratio      Ratio   turnover  Rate     Rate  _algorithm_  _status_

  1  89.559  1.11662   0.054843  0.89556 0.94664 0.85528  __-SS-GA    SUCCESSFUL
  2  18.932  0.37676   0.029401  2.65424 0.97060 0.84850  BR-BS-GA    SUCCESSFUL
  3  30.405  0.93095   0.030055  1.07418 0.97027 0.91984  FR-BS-GA    SUCCESSFUL
  4 229.379  0.76705   0.090748  1.30370 0.91309 0.79612  __-SS-GA    SUCCESSFUL
  5  71.268  1.90048   0.029686  0.52618 0.97255 0.95194  BR-BS-GA    SUCCESSFUL
  6  36.043  0.77313   0.028998  1.29345 0.97122 0.90423  FR-BS-GA    SUCCESSFUL
  7  68.019  1.14994   0.030353  0.86961 0.97045 0.92772  FR-BS-GA    SUCCESSFUL
  8 432.363  1.19969   0.058859  0.83355 0.95131 0.88719  __-SS-GA    SUCCESSFUL
  9  49.399  0.84012   0.030331  1.19030 0.96999 0.91296  BR-BS-GA    SUCCESSFUL
 10 138.232  1.73549   0.032036  0.57621 0.97012 0.94567  FR-BS-GA    SUCCESSFUL
 11  68.573  0.92867   0.030176  1.07681 0.97024 0.91947  FR-BS-GA    SUCCESSFUL
 12  71.798  1.02480   0.030961  0.97580 0.96964 0.92035  FR-BS-GA    SUCCESSFUL
 13 176.761  2.34866   0.051398  0.42577 0.95939 0.93192  __-SS-GA    SUCCESSFUL
 14 121.846  1.37835   0.031072  0.72550 0.97021 0.93966  FR-BS-GA    SUCCESSFUL
```

Example 2.7. Two-Echelon System: Penalty Costs

In this example, assume that estimates of backorder penalty costs are known for the problem in Example 2.6. Rather than using service level heuristics, as in Example 2.6, a penalty cost heuristic is used to calculate inventory policies.

The penalty costs are given in the following data set. There are no penalty costs for backorders at the warehouses, since customers are only seen at the retailers.

```
data pcosts;
   format warehouse $3. retailer $3. sku $3. color $4. ;
   input sku $ color $ warehouse $ retailer $ penaltyCost;
   datalines;
M01  BLUE  W01     .     .
M01  BLUE  W01  R01   6.7
M01  BLUE  W01  R02  10.2
M01  RED   W01     .     .
M01  RED   W01  R01   8.4
M01  RED   W01  R03   5.6
M01  RED   W01  R04   9.1
M02  BLUE  W02     .     .
M02  BLUE  W02  R01   3.4
M02  BLUE  W02  R02   6.9
M02  BLUE  W02  R03   7.7
M02  BLUE  W02  R04  12.4
M02  RED   W02     .     .
M02  RED   W02  R03   7.5
;
```

This data set is merged with in6_fr to produce the input data set in7. The serviceLevel and serviceType variables are dropped from the in6_fr data set, as these will not be needed when using penalty costs. However, if these variables are left in the data set, they will simply be ignored when the METHOD= option is specified as PENALTY.

```
data in7;
   merge in6_fr (drop=serviceLevel serviceType)
         pcosts;
   by sku color warehouse retailer;
run;
```

The call to PROC IRP is as follows. There are two main differences between this call to PROC IRP and the call in Example 2.6. First, METHOD=PENALTY is specified on the PROC IRP statement, to indicate that a penalty cost heuristic should be used to compute the policies. Also, the SERVICE statement is removed and the PENALTY statement is added to specify the variable in the input data set that gives the penalty costs.

```
proc irp data=in7 out=out7 method=penalty;
   holdingcost holdingCost;
   itemid sku color warehouse;
   leadtime / mean=LTmean variance=LTvar;
   location retailer;
   penalty / cost=penaltyCost;
   policytype policyType;
   replenishment / fcost=fixedCost;
   reviewtimedemand / mean=RTDmean variance=RTDvar;
run;
```

The output data set from this call to PROC IRP is shown in Output 2.7.1.

Output 2.7.1. Inventory Policies Using Penalty Cost Heuristic

PROC IRP Results for Two-Echelon System
Penalty Cost Heuristic

Obs	sku	color	warehouse	retailer	reorder Level	order UpTo Level	avg Inventory	avg Backorder	avg Order Freq
1	M01	BLUE	W01		152	379	108.314	3.7547	0.32990
2	M01	BLUE	W01	R01	231	232	29.470	1.1058	1.00000
3	M01	BLUE	W01	R02	62	63	17.201	1.1063	1.00000
4	M01	RED	W01		302	634	153.342	13.5790	0.42581
5	M01	RED	W01	R01	227	228	76.872	2.6729	1.00000
6	M01	RED	W01	R03	192	193	24.167	2.3598	1.00000
7	M01	RED	W01	R04	383	384	106.581	2.4984	1.00000
8	M02	BLUE	W02		664	1258	392.298	19.2468	0.42808
9	M02	BLUE	W02	R01	216	217	49.596	5.5738	1.00000
10	M02	BLUE	W02	R02	235	236	61.901	6.3973	1.00000
11	M02	BLUE	W02	R03	197	198	55.122	3.3288	1.00000
12	M02	BLUE	W02	R04	481	482	138.297	1.9926	1.00000
13	M02	RED	W02		187	490	195.916	4.6490	0.23837
14	M02	RED	W02	R03	253	254	85.205	5.7533	1.00000

Obs	avgCost	inventory Ratio	backorder Ratio	turnover	fill Rate	ready Rate	_algorithm_	_status_
1	107.007	1.20349	0.04172	0.83091	0.95925	0.88183	__-SS-GA	SUCCESSFUL
2	29.511	0.43984	0.01651	2.27353	0.98350	0.90262	PC-BS-GA	SUCCESSFUL
3	35.710	0.74788	0.04810	1.33711	0.95243	0.87593	PC-BS-GA	SUCCESSFUL
4	268.804	0.83793	0.07420	1.19341	0.92862	0.82502	__-SS-GA	SUCCESSFUL
5	80.107	1.53745	0.05346	0.65043	0.95111	0.91718	PC-BS-GA	SUCCESSFUL
6	40.040	0.57541	0.05618	1.73789	0.94439	0.82985	PC-BS-GA	SUCCESSFUL
7	92.014	1.17122	0.02746	0.85381	0.97319	0.93320	PC-BS-GA	SUCCESSFUL
8	479.147	1.19969	0.05886	0.83355	0.95131	0.88719	__-SS-GA	SUCCESSFUL
9	53.668	0.59043	0.06635	1.69368	0.93466	0.82710	PC-BS-GA	SUCCESSFUL
10	127.708	1.04917	0.10843	0.95313	0.90223	0.83795	PC-BS-GA	SUCCESSFUL
11	82.958	0.77636	0.04688	1.28806	0.95390	0.88071	PC-BS-GA	SUCCESSFUL
12	110.452	1.22386	0.01763	0.81708	0.98264	0.95192	PC-BS-GA	SUCCESSFUL
13	186.045	2.30490	0.05469	0.43386	0.95681	0.92798	__-SS-GA	SUCCESSFUL
14	131.763	1.00241	0.06769	0.99759	0.93582	0.87737	PC-BS-GA	SUCCESSFUL

Statement and Option Cross-Reference Tables

The following tables reference the statements and options in the IRP procedure that are illustrated by the examples in this section.

Table 2.20. Statements Specified in Examples

Statement	Examples						
	1	**2**	**3**	**4**	**5**	**6**	**7**
HOLDINGCOST	X	X	X		X	X	X
ITEMID	X	X	X	X	X	X	X
LEADTIME	X	X	X		X	X	X
LEADTIMEDEMAND				X			
LOCATION						X	X
PENALTY		X	X				X
POLICYTYPE	X	X	X			X	X
REPLENISHMENT	X	X	X		X	X	X
REVIEWTIMEDEMAND	X	X	X	X	X	X	X
SERVICE	X				X	X	

Table 2.21. Options Specified in Examples

Option	Examples						
	1	**2**	**3**	**4**	**5**	**6**	**7**
COST=		X	X				X
DATA=	X	X	X	X	X	X	X
FCOST=	X	X	X		X	X	X
LEVEL=	X				X	X	
LOTSIZE=	X	X	X				
MAXFREQ=	X	X	X				
MEAN= (LEADTIME)	X	X	X		X	X	X
MEAN= (LEADTIMEDEMAND)				X			
MEAN= (REVIEWTIMEDEMAND)	X	X	X	X	X	X	X
METHOD=	X	X	X		X	X	X
MINSIZE=	X	X	X				
OPTIMAL			X				
OUT=	X	X	X	X	X	X	X
TYPE=	X					X	
VARIANCE= (LEADTIME)	X	X	X			X	X
VARIANCE= (LEADTIMEDEMAND)				X			
VARIANCE= (REVIEWTIMEDEMAND)	X	X	X	X	X	X	X

References

Ehrhardt, R. and Mosier, C. (1984), "A Revision of the Power Approximation for Computing (s, S) Policies," *Management Science*, 30, 618–622.

Graves, S. C., Rinnooy Kan, A. H. G., and Zipkin, P. H., eds. (1993), *Handbooks in Operations Research and Management Science, 4: Logistics of Production and Inventory*, Netherlands: North-Holland.

Matta, K. F. and Sinha, D. (1995), "Policy and Cost Approximations of Two-Echelon Distribution Systems with a Procurement Cost at the Higher Echelon," *IIE Transactions*, 27, 638–645.

Schneider, H. (1978), "Methods for Determining the Re-order Point of an (s, S) Ordering Policy When a Service Level Is Specified," *Journal of the Operational Research Society*, 29, 1181–1193.

Schneider, H. (1981), "Effects of Service-Levels on Order-Points or Order-Levels in Inventory Models," *International Journal of Production Research*, 19, 615–631.

Schneider, H. and Ringuest, J. L. (1990), "Power Approximation for Computing (s, S) Policies Using Service Level," *Management Science*, 36, 822–834.

Silver, E. A., Pyke, D. F., and Peterson, R. (1998), *Inventory Management and Production Planning and Scheduling*, New York: John Wiley and Sons, Inc.

Svoronos, A. and Zipkin, P. H. (1991), "Evaluation of One-for-One Replenishment Policies for Multi-Echelon Inventory Systems," *Management Science*, 37, 68–83.

Tijms, H. and Groenevelt, H. (1984), "Simple Approximations for the Reorder Point in Periodic and Continuous Review (s, S) Inventory Systems with Service Level Constraints," *European Journal of Operations Research*, 17, 175–190.

Zheng, Y. and Chen, F. (1992), "Inventory Policies with Quantized Ordering," *Naval Research Logistics*, 39, 285–305.

Zheng, Y. and Federgruen, A. (1992), "Finding Optimal (s, S) Policies Is About as Simple as Evaluating a Single Policy," *Operations Research*, 39, 654–665.

Zipkin, P. H. (2000), *Foundations of Inventory Management*, New York: McGraw-Hill.

Chapter 3

The MIRP Procedure (Experimental)

Chapter Contents

Chapter 3
The MIRP Procedure (Experimental)

Overview

The MIRP procedure calculates inventory control parameters for all stock-keeping locations in general supply chain networks, which may have any combination of serial, assembly, and distribution subnetworks. These parameters ensure that service requirements are satisfied at minimum inventory costs for every supply chain. PROC MIRP currently supports the base-stock replenishment policy and the ready rate as the service level requirement.

Getting Started

Before you create data sets for PROC MIRP, you must define the *base period*. It could be any time unit, such as one day, three days, two weeks, or one month. Only one replenishment order is allowed per base period. The mean and variance of demand per base period are provided in the input data sets. Lead times from outside suppliers or between locations must be integral multiples of the base period. For simplicity, the term *period* refers to the base period.

Single-Echelon Network

Consider a grocery store in a local community. One of its stock-keeping units (SKUs) faces a stable demand with a mean of 10 units and a variance of 9 units per day. On a daily basis, the store checks the SKU's inventory level and places necessary replenishment orders to an external supplier. If an order is placed, it takes one day to receive a shipment. The store would like to keep a 95% service level (i.e., ready rate) for the SKU. Any unsold units at the end of a day cost $1 to hold per day per unit. Unsatisfied demand is backlogged and fulfilled as soon as new shipments arrive.

Figure 3.1 illustrates this example graphically. A circle (or *node*) is used to represent the SKU and the location that keeps it; for this example, the node is labeled RW to indicate SKU R at location W. See the "Labeling of a Supply Chain" section on page 81 for more details about labeling nodes in a supply chain. In addition, L stands for the lead time from the external supplier, and h represents the unit holding cost. The pair $(10, 9)$ gives the mean and variance of the daily demand. Similar notations are used throughout this chapter. The base period is defined as one day.

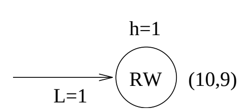


Figure 3.1. Single-Echelon Network

The store can use PROC MIRP to determine the optimal inventory level for the SKU such that the service level requirement is met with a minimum total inventory cost. PROC MIRP requires three data sets as input. For the example in Figure 3.1, these data sets are as follows.

```
data nodedata;
   input NETWORKID $ ITEMID $ DESCRIPTION $15.
         LEADTIME SERVICELEVEL HOLDINGCOST;
   datalines;
N1 RW single location 1 0.95 1
;

data arcdata;
   input NETWORKID $ HEAD $ TAIL $ QTY;
   datalines;
N1 EXTERNAL RW 1
;

data demanddata;
   input NETWORKID $ ITEMID $ MEAN VARIANCE;
   datalines;
N1 RW 10 9
;
```

The NETWORKID variable specifies the name of the network to which an SKU belongs. This variable enables users to determine inventory parameters for multiple independent networks with a single call to the procedure. The arcdata data set contains the bill of material information on each arc. It is mainly used in the assembly process. For more information, see the "Two-Echelon Assembly Network" section on page 73. Notice that 'EXTERNAL' is a keyword reserved for external suppliers, and it must be capitalized. If an SKU is purchased from an outside supplier, an arc from 'EXTERNAL' to it must be included in the data set. The demanddata data set contains the mean and variance of demand for each customer-facing SKU. An SKU is customer-facing if it receives demand directly from end customers, not from other locations within the network. An SKU could receive demand externally as well as internally. In that case, the data set must provide information about the external demand. See the "Syntax" section on page 77 for a more detailed description of the data sets and the definition of the variables.

Once the three data sets are created, the following call to PROC MIRP computes the optimal inventory level for the SKU.

```
proc mirp NODEDATA=nodedata ARCDATA=arcdata
          DEMANDDATA=demanddata NETWORKCNT=1;
run;
```

The NETWORKCNT option specifies the number of networks in the input data sets. The name of the output data set is not specified, so the procedure uses the default name as determined by the SAS system. Figure 3.2 displays the resulting output data set.

Obs	NETWORKID	ITEMID	DESCRIPTION	REORDERPOINT	DEMANDMEAN	DEMANDVAR	ORDERMEAN	ORDERVAR	BKLGUPMEAN	BKLGUPVAR	BKLGDOWNMEAN	BKLGDOWNVAR	ONHANDMEAN	ONHANDVAR	OHCOST	PSCOST	SLREQ	SLSUG
1	N1	RW	single location	28	10	9	10	9	0	0	0.17078	0.83164	8.12205	16.3297	8.12205	10	0.95	0.96045

Figure 3.2. Output Data Set of Single-Echelon Network

The variables in the output data set are defined in the "OUT= Data Set" section on page 81. It is worthwhile to point out that the optimal reorder point of the SKU is 28. To use this number, the store needs to calculate its inventory position, which is the sum of on-hand inventory and outstanding orders (shipments yet to be received) minus any backlogs. If it is below 28, the store should then order the difference such that its inventory position is raised up to 28.

Two-Echelon Distribution Network

Consider a simple distribution network with one warehouse and two retailer locations. A finished good F is stored at the warehouse W with a lead time of 3 weeks from an external supplier, and a unit holding cost of $16 per week. The finished good is shipped to the two retailers C and D. The unit holding costs at C and D are $18 per week, higher than that of W. In general, the unit holding costs increase (or at least do not decrease) as SKUs move downstream. This is simply because the value of a product tends to rise as it moves down toward the end of a supply chain. The lead time between W and C is 3 weeks, and between W and D is 4 weeks. The demand at C from end customers has a mean of 20 and a variance of 25 per week. The demand at D has a mean of 10 and a variance of 9 per week. The minimum required service levels are 95% for both C and D, and 50% for W. Figure 3.3 illustrates this distribution network with notations similar to the "Single-Echelon Network" section on page 69. In this example, the base period is defined as one week.

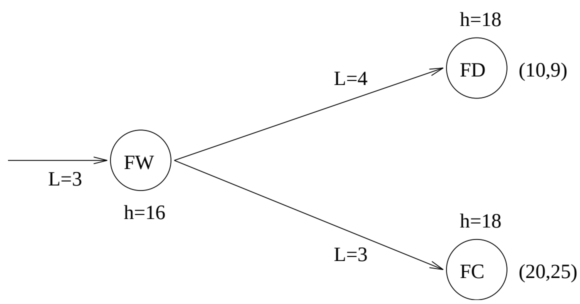


Figure 3.3. Two-Echelon Distribution Network

There are three nodes in the network. Each node is labeled by the name of the finished good and the location where the good is stored. For example, the node FW stands for the finished good F at the warehouse W. For more information about such labeling of nodes in a supply chain, see the "Labeling of a Supply Chain" section on page 81.

PROC MIRP can be used to minimize the inventory cost for the entire network while meeting the service level constraints at all locations.

```
data nodedata;
   input NETWORKID $ ITEMID $ DESCRIPTION $15.
         LEADTIME SERVICELEVEL HOLDINGCOST;
   datalines;
N2 FW warehouse       3 0.5  16
N2 FC retailer 1      3 0.95 18
N2 FD retailer 2      4 0.95 18
;

data arcdata;
   input NETWORKID $ HEAD $ TAIL$ QTY;
   datalines;
N2 EXTERNAL FW 1
N2 FW       FC 1
N2 FW       FD 1
;

data demanddata;
   input NETWORKID $ ITEMID $ MEAN VARIANCE;
   datalines;
N2 FC 20 25
N2 FD 10 9
;


proc mirp NODEDATA=nodedata ARCDATA=arcdata
          DEMANDDATA=demanddata NETWORKCNT=1;
run;
```

In the arcdata data set, QTY=1 between FW and FC means one unit of F shipped from the warehouse W is received as one unit of F at retailer C, without any loss. If there is any loss during the transportation, say 10%, then QTY would be set to the value of 0.9. Other types of yields can be modeled in a similar fashion. In the output data set shown in Figure 3.4, reorder points are given for each node. Note that the suggested service level (SLSUG) for FW is 56.9%, which is higher than the required service level of 50%. This is because the total inventory cost of the network is lower when FW is operated at the suggested service level. Example 3.2 on page 90 shows how the service level requirements affect the total inventory investment of a network.

Obs	NETWORKID	ITEMID	DESCRIPTION	REORDERPOINT	DEMANDMEAN	DEMANDVAR	ORDERMEAN	ORDERVAR	BKLGUPMEAN	BKLGUPVAR	BKLGDOWNMEAN
1	N2	FC	retailer 1	101	20	25	20	25	2.48353	13.7535	0.32476
2	N2	FD	retailer 2	66	10	9	10	9	2.48353	13.7535	0.23276
3	N2	FW	warehouse	121	30	34	30	34	0.00000	0.0000	4.96707

Obs	BKLGDOWNVAR	ONHANDMEAN	ONHANDVAR	OHCOST	PSCOST	SLREQ	SLSUG
1	3.2648	18.7995	101.536	338.391	1080	0.95	0.95299
2	1.7538	13.7241	52.310	247.034	720	0.95	0.95498
3	55.0139	5.1929	46.229	83.086	1440	0.50	0.56910

Figure 3.4. Output Data Set of Two-Echelon Distribution Network

Two-Echelon Assembly Network

Consider a manufacturing process in a plant P. Two components, A and B, are assembled into a finished good, F. Both components are outsourced from external suppliers. The lead times are 1 period and 2 periods for A and B, respectively. The unit holding cost is \$1 per period for A to be stored at the plant, and \$2 for B. It takes three units of A and two units of B to produce one unit of F. The assembly process takes 3 periods. It costs the plant \$13 per period to store one unit of F. The minimum service level required is 95% for F, and 50% for both A and B. The finished good F

faces customer demand with a mean of 16 units and a variance of 36 units per period. This assembly process is illustrated in Figure 3.5.

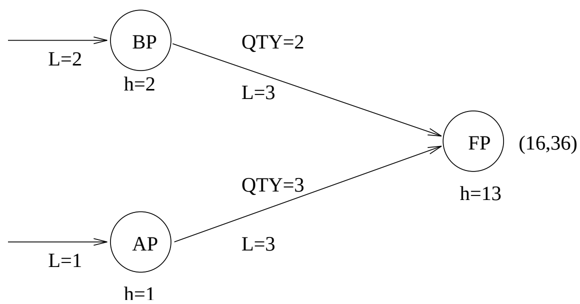


Figure 3.5. Two-Echelon Assembly Network

Similar to the previous examples, the labels of the nodes in the network give the SKU (i.e., the components or the finished good) and the location (i.e., the plant). See the "Labeling of a Supply Chain" section on page 81 for details.

The total holding cost of the two components that go into the finished good can be computed as follows.

$$\begin{aligned} & \text{unit holding cost of A} \times \text{QTY of A into F} \\ + & \text{ unit holding cost of B} \times \text{QTY of B into F} \\ = & \ 1 \times 3 + 2 \times 2 = 7. \end{aligned}$$

Since the unit holding cost of F is \$13, a significant value is added during the assembly process. It also implies that it is much more expensive to hold the finished good than the components.

```
data nodedata;
   input NETWORKID $ ITEMID $ DESCRIPTION $15.
         LEADTIME SERVICELEVEL HOLDINGCOST;
   datalines;
N3 AP component 1      1 0.5  1
N3 BP component 2      2 0.5  2
N3 FP finished good    3 0.95 3
;

data arcdata;
   input NETWORKID $ HEAD $ TAIL $ QTY;
   datalines;
N3 EXTERNAL AP 1
N3 EXTERNAL BP 1
N3 AP       FP 3
N3 BP       FP 2
;
```

```
data demanddata;
   input NETWORKID $ ITEMID $ MEAN VARIANCE;
   datalines;
N3 FP 16 36
;


proc mirp NODEDATA=nodedata ARCDATA=arcdata
          DEMANDDATA=demanddata NETWORKCNT=1;
run;
```

Notice that in the arcdata data set, the bill of material relationship (i.e., QTY) is assigned between the components and the finished good. The output data set is shown in Figure 3.6.

Obs	NETWORKID	ITEMID	DESCRIPTION	REORDERPOINT	DEMANDMEAN	DEMANDVAR	ORDERMEAN	ORDERVAR	BKLGUPMEAN	BKLGUPVAR	BKLGDOWNMEAN
1	N3	AP	component 1	96	48	324	48	324	0.00000	0.0000	12.6943
2	N3	BP	component 2	96	32	144	32	144	0.00000	0.0000	9.8628
3	N3	FP	finished good	95	16	36	16	36	6.26050	50.5155	0.4668

Obs	BKLGDOWNVAR	ONHANDMEAN	ONHANDVAR	OHCOST	PSCOST	SLREQ	SLSUG
1	335.724	10.1972	177.467	10.1972	48	0.50	0.54559
2	207.086	8.3227	122.614	16.6453	128	0.50	0.54139
3	6.605	25.1316	172.555	75.3948	144	0.95	0.95231

Figure 3.6. Output Data Set of Two-Echelon Assembly Network

You could also solve all three of the previous examples in a single call to PROC MIRP by combining the input data as follows. Figure 3.7 gives the output data set from this call to PROC MIRP.

```
data nodedata;
   input NETWORKID $ ITEMID $ DESCRIPTION $15.
         LEADTIME SERVICELEVEL HOLDINGCOST;
   datalines;
N1 RW single location 1 0.95 1
N2 FW warehouse       3 0.5  16
N2 FC retailer 1      3 0.95 18
N2 FD retailer 2      4 0.95 18
N3 AP component 1     1 0.5  1
N3 BP component 2     2 0.5  2
N3 FP finished good   3 0.95 3
;

data arcdata;
   input NETWORKID $ HEAD $ TAIL $ QTY;
   datalines;
N1 EXTERNAL RW 1
N2 EXTERNAL FW 1
N2 FW       FC 1
N2 FW       FD 1
N3 EXTERNAL AP 1
N3 EXTERNAL BP 1
N3 AP       FP 3
N3 BP       FP 2
;

data demanddata;
   input NETWORKID $ ITEMID $ MEAN VARIANCE;
   datalines;
N1 RW 10 9
N2 FC 20 25
N2 FD 10 9
N3 FP 16 36
;


proc mirp NODEDATA=nodedata ARCDATA=arcdata
          DEMANDDATA=demanddata NETWORKCNT=3;
run;
```

Notice that distinct network IDs are assigned to each network. Since PROC MIRP computes inventory control parameters one network at a time, items and arcs in the data sets must be grouped by the networks that they belong to. In addition, the sequence of networks in the three input data sets must be the same. In all the data sets of the above example, networks are arranged in the same order of N1, N2, and N3.

Obs	NETWORKID	ITEMID	DESCRIPTION	REORDERPOINT	DEMANDMEAN	DEMANDVAR	ORDERMEAN	ORDERVAR	BKLGUPMEAN	BKLGUPVAR	BKLGDOWNMEAN
1	N1	RW	single location	28	10	9	10	9	0.00000	0.0000	0.1708
2	N2	FC	retailer 1	101	20	25	20	25	2.48353	13.7535	0.3248
3	N2	FD	retailer 2	66	10	9	10	9	2.48353	13.7535	0.2328
4	N2	FW	warehouse	121	30	34	30	34	0.00000	0.0000	4.9671
5	N3	AP	component 1	96	48	324	48	324	0.00000	0.0000	12.6943
6	N3	BP	component 2	96	32	144	32	144	0.00000	0.0000	9.8628
7	N3	FP	finished good	95	16	36	16	36	6.26050	50.5155	0.4668

Obs	BKLGDOWNVAR	ONHANDMEAN	ONHANDVAR	OHCOST	PSCOST	SLREQ	SLSUG
1	0.832	8.1221	16.330	8.122	10	0.95	0.96045
2	3.265	18.7995	101.536	338.391	1080	0.95	0.95299
3	1.754	13.7241	52.310	247.034	720	0.95	0.95498
4	55.014	5.1929	46.229	83.086	1440	0.50	0.56910
5	335.724	10.1972	177.467	10.197	48	0.50	0.54559
6	207.086	8.3227	122.614	16.645	128	0.50	0.54139
7	6.605	25.1316	172.555	75.395	144	0.95	0.95231

Figure 3.7. Output Data Set for Three Networks

Syntax

The following statements are used in PROC MIRP.

PROC MIRP *options* ;
  NODE / *node options* ;
  ARC / *arc options* ;
  DEMAND / *demand options* ;

PROC MIRP Statement

PROC MIRP *options* **;**

The following options can appear on the PROC MIRP statement. The NODEDATA=, ARCDATA=, and DEMANDDATA= data sets are required, while the NETWORKCNT= option and the OUT= data set are optional.

NODEDATA=*SAS-data-set*
names the SAS data set that contains the information about each item (or node) in the networks to be analyzed. Required information includes network ID, item ID, item description, lead time, minimum service level, and unit holding cost. The NODEDATA= data set is required and must be grouped by the NETWORKID= variable.

ARCDATA=*SAS-data-set*
names the SAS data set that contains information about the arcs in the networks to be analyzed. Required information includes network ID, head item ID, tail item ID, and quantity of the head item needed to produce one unit of the tail item. The ARCDATA= data set is required and must be grouped by the NETWORKID= variable.

DEMANDDATA=*SAS-data-set*
names the SAS data set that contains information about external demand at all customer-facing items in the networks to be analyzed. Required information includes network ID, item ID, mean demand, and variance of demand. The DEMANDDATA= data set is required and must be grouped by the NETWORKID= variable.

NETWORKCNT=*networkcnt*
specifies the maximum number of networks to be analyzed by PROC MIRP. If the NETWORKCNT= option is omitted, a default value of 20 is used. If the input data sets contain more networks than that specified by NETWORKCNT= option, the extra networks are ignored.

When PROC MIRP is used to process multiple networks, the data of the networks must be grouped in the same sequence in all three of the input data sets. For example, if the NODEDATA= data set contains networks in the order of N1, N2, and N3, then the ARCDATA= and DEMANDDATA= data sets must group their networks in the same way.

OUT=*SAS-data-set*
names the output data set that contains inventory control parameters, suggested service levels, and other inventory metrics for all items in all networks. This data set also includes network ID, item ID, and descriptions specified in the input data sets. If the OUT= is omitted, the SAS system creates a data set and names it according to the DATA*n* naming convention. See the "OUT= Data Set" section on page 81 for more details on the variables in this data set.

NODE Statement

NODE */ node options* **;**

The NODE statement specifies the variables contained in the NODEDATA= data set. The variables are required. If any of the following statements is omitted, PROC MIRP looks for the default name of the variable in the NODEDATA= data set.

NETWORKID=*variable*
identifies the variable in the NODEDATA= data set that specifies the network ID for each item to be analyzed. The type of the NETWORKID= variable must be character. The default name is NETWORKID.

ITEMID=*variable*
identifies the variable in the NODEDATA= data set that specifies the ID for each item to be analyzed. The type of the ITEMID= variable must be character. The default name is ITEMID.

DESCRIPTION=*variable*
identifies the variable in the NODEDATA= data set that specifies the description for each item to be analyzed. The type of the DESCRIPTION= variable must be character. The default name is DESCRIPTION.

LEADTIME=*variable*
identifies the variable in the NODEDATA= data set that specifies the lead time for each item to be analyzed. The variable must be numeric, and the value must be an integral multiple of the base period. The default name is LEADTIME.

SERVICELEVEL=*variable*
identifies the variable in the NODEDATA= data set that specifies the minimum service level requirement for each item to be analyzed. The variable must be numeric, and the value must be between 0 and 0.9999. The default name is SERVICELEVEL.

HOLDINGCOST=*variable*
identifies the variable in the NODEDATA= data set that specifies the unit holding cost per period for each item to be analyzed. The variable must be numeric, and the value must be nonnegative. The default name is HOLDINGCOST.

ARC Statement

ARC */ arc options* **;**

The ARC statement identifies the variables contained in the ARCDATA= data set. These variables are required. If any of the following statements is omitted, PROC MIRP looks for the default name of the variable in the ARCDATA= data set.

NETWORKID=*variable*
identifies the variable in the ARCDATA= data set that specifies the network ID for each arc to be analyzed. The type of the variable must be character, and the values must be consistent with those in the NODEDATA= data set. The default name is NETWORKID.

HEAD=*variable*
identifies the variable in the ARCDATA= data set that specifies the ITEMID of the head of each arc to be analyzed. The head of an arc is the *predecessor* node of the arc. The type of the variable must be character, and the values must be consistent with those in the NODEDATA= data set. The default name is HEAD.

TAIL=*variable*
identifies the variable in the ARCDATA= data set that specifies the ITEMID of the tail of each arc to be analyzed. The tail of an arc is the *successor* node of the arc. The type of the variable must be character, and the values must be consistent with those in the NODEDATA= data set. The default name is TAIL.

QTY=*variable*
identifies the variable in the ARCDATA= data set that specifies the bill of material (BOM) quantity between the head and the tail items of each arc. It is the number of units of the head item required to produce one unit of the tail item. The variable must be numeric, and the value must be positive. The default name is QTY.

DEMAND Statement

DEMAND */ demand options* **;**

The DEMAND statement identifies the variables contained in the DEMANDDATA= data set. These variables are required. If any of the following statements is omitted, PROC MIRP looks for the default name of the variable in the DEMANDDATA= data set.

NETWORKID=*variable*
identifies the variable in the DEMANDDATA= data set that specifies the NETWORKID for each item to be analyzed. The type of the NETWORKID= variable must be character, and the values must be consistent with those in the NODEDATA= data set. The default name is NETWORKID.

ITEMID=*variable*
identifies the variable in the DEMANDDATA= data set that specifies the ID for each item to be analyzed. The type of the variable must be character, and the values must be consistent with those in the NODEDATA= data set. The default name is ITEMID.

MEAN=*variable*
identifies the variable in the DEMANDDATA= data set that specifies the mean of the demand of each customer-facing item to be analyzed. The variable must be numeric, and the values must be positive. The default name is MEAN.

VARIANCE=*variable*
identifies the variable in the DEMANDDATA= data set that specifies the variance of demand of each customer-facing item to be analyzed. The variable must be numeric, and the values must be positive. The default name is VARIANCE.

Details

Labeling of a Supply Chain

In general, many SKUs are supported by one physical supply chain. One location in the network could hold many different SKUs, while one SKU could be held in many different locations. In order to use PROC MIRP, you must map the physical network to a "virtual" one whose nodes are uniquely defined by the pair of SKU and location. The following two examples demonstrate how this mapping works. Example 3.1 on page 85 illustrates how to label a more complex supply chain.

A product is held in multiple physical locations.

A product F is distributed from a warehouse W to three locations, A, B, and C. In order to determine the inventory control parameters for product F at all four locations, four nodes must be defined in the "virtual" supply chain, and they must be uniquely labeled. The labeling convention used throughout this chapter labels the nodes by their SKU and location, such as FW for product F at the warehouse W, FA for product F at the location A, etc.

A physical location holds different products.

Three components, D, E, and F, are assembled into a finished good G in a manufacturing plant P. In order to determine the inventory control parameters for the components as well as the finished good, a four-node network must be created to model the assembly process. The labels DP, EP, and FP are used for the components D, E, and F at the plant P, respectively. The label GP is used for the finished good G at the plant P. Note that four nodes are necessary to model this process even though the process occurs at one single location.

Inventory Components

Two inventory components are considered in PROC MIRP: the on-hand inventory and the pipeline inventory. The on-hand inventory is the stock available at a location that can be used to meet demand immediately. The pipeline inventory is the stock in transit from the supply node (or the external supplier) to the location. It becomes on-hand inventory once it is received at the location.

OUT= Data Set

The OUT= data set contains the inventory control parameters and inventory measures for all nodes in input networks that are successfully processed. Recall that a node is uniquely defined by the SKU and location in a network. If an error occurs while a network is being processed, an error message is written to the log file.

Variables in the OUT= Data Set

For the rest of the chapter, "item" and "node" are used interchangeably.

NETWORKID
contains the network identification of a node.

ITEMID
contains the identification of a node.

DESCRIPTION
contains the descriptive information for a node.

REORDERPOINT
contains the target replenishment level for a node under a base-stock inventory replenishment policy. When the inventory position of a node drops below its reorder point at any review period, a replenishment order is made to raise the inventory position up to the reorder point. For more information about the policy, refer to the "Replenishment Policy" section on page 84.

DEMANDMEAN
contains the average demand received by a node in each period. If a node faces external demand only, the average demand is the same as the mean of the demand specified in the DEMANDDATA= data set. Otherwise, it is derived by PROC MIRP.

DEMANDVAR
contains the variance of demand received by a node in each period. If a node faces external demand only, the variance is the same as the variance of the demand specified in the DEMANDDATA= data set. Otherwise, it is derived by PROC MIRP.

ORDERMEAN
contains the average amount a node orders from its immediate predecessors at each period.

ORDERVAR
contains the variance of the amount a node orders from its immediate predecessors at each period.

BKLGUPMEAN
contains the average backlog experienced by a node from its immediate predecessors at each period.

BKLGUPVAR
contains the variance of the backlogs experienced by a node from its immediate predecessors at each period.

BKLGDOWNMEAN
contains the average backlog that a node has at each period.

BKLGDOWNVAR
contains the variance of the backlog that a node has at each period.

ONHANDMEAN
contains the average on-hand inventory that a node has at each period.

ONHANDVAR
contains the variance of the on-hand inventory that a node has at each period.

OHCOST
contains the average cost of the on-hand inventory that a node has at each period.

PSCOST
contains the average cost of the pipeline inventory that a node has at each period. The pipeline inventory are the stocks in transit to the node.

SLREQ
contains the minimum required service level of a node. This is the same as in the value of SERVICELEVEL= variable in the NODEDATA= data set.

SLSUG
contains the service level suggested by PROC MIRP for a node. When a node faces external demand (i.e., customer-facing), the suggested service level is equal to or slightly higher than the required service level. For nodes facing internal demand (i.e., demand from nodes internal to the network), PROC MIRP optimizes their service levels based on the cost structure of the network. The suggested service levels for these nodes may be much higher than the required service levels.

Model Description

This section describes the key concepts upon which PROC MIRP is based.

Assumptions

The underlying assumptions of PROC MIRP are as follows:

- The holding cost is linear in the amount of inventory held. That is, the holding cost is equal to the amount of inventory held multiplied by the unit holding cost.
- There is no fixed ordering cost. When such a cost exists, the base-stock control policy is no longer optimal. An (s, S) inventory control policy (also called min-max) should be used instead. This policy will be supported in a future release.
- Lead times are deterministic and are integral multiples of the base period. In practice, there are two major components of lead time: fulfillment-related lead time and transit-related lead time. When the on-hand inventory at a location is insufficient to satisfy orders from downstream, a backlog occurs and only a fraction of orders are shipped immediately. The remaining portion is filled in some future period, which is likely to be random. This is called *fulfillment-related* lead time. The time to ship an order (full or partial) from one location to another is called *transit-related* lead time, which is more or less constant. In the multi-echelon inventory theory, only the transit-related lead time is considered as the lead time, while the fulfillment-related lead time is taken into account implicitly by the algorithm (more specifically, by incorporating backlogs into the calculation of control parameters).

- The orders between locations within a network and the external demand from end customers occur only once in a review period. Since PROC MIRP is used mainly for the inventory planning at the strategic level, this is a reasonable assumption. It is simplistic when managing inventory at the store level because the external demand most likely comes in continuously in small amounts instead of one big demand in a period. As long as the service level is measured as the non-stockout probability at the end of a period, or the percentage of total demand being satisfied in a period, the inventory control parameters are the same on either assumption of the demand, whether it is continuous or discrete in time.
- The external demand is integral-valued, independent between periods, and identically distributed (stationary). Future releases will allow nonstationary demand as input.

Replenishment Policy

The current version of PROC MIRP supports one replenishment policy, the base-stock policy. More specifically, when the inventory position falls to or below the reorder level, R, an order is placed so as to bring the inventory position up to R.

Service Level

Service levels are often used to evaluate the effectiveness of an inventory replenishment policy. PROC MIRP currently supports the use of the ready rate, which is also called non-stockout probability in the inventory literature. It is the probability of having no stockout in a review period.

Error Processing

This section describes the errors that could occur when running PROC MIRP. The errors could be caused by incorrect syntax or problems with the input data. If an error occurs, PROC MIRP always writes an error or warning message to the log file. PROC MIRP also tries to fix the problem and continue. If the problem cannot be resolved, MIRP PROC exits.

If any of the following errors occurs, MIRP PROC exits immediately:

- Any of the input data sets does not exist or is empty.
- Any of the required variables in the input data sets does not exist.
- The number of networks in the input data sets are different.
- The input data sets do not have the same network IDs, or the IDs are not in the same sequence.
- NETWORKCNT is zero or negative.

If any of the following errors occurs, PROC MIRP skips the current network being processed and moves on to the next network:

- A head or tail item of the current network in the ARCDATA= data set is not defined in the NODEDATA= data set.
- An item of the current network in the DEMANDDATA= data set is not defined in the NODEDATA= data set.
- The head and tail items of an arc in the current network are identical.
- A cycle exists in the current network.
- The mean or variance of demand of an item is zero, negative, or missing.

If any of the following errors occur, PROC MIRP fixes the error and continues:

- The required service level of an item exceeds 0.9999. It is reset to 0.9999.
- The required service level of an item is missing or negative. It is reset to 0.5.
- The unit holding cost of an item is missing or negative. It is reset to 0.
- The lead time of an item is missing or negative. It is reset to 0.
- The lead time of an item is not integer. It is rounded down to the nearest integer.
- The QTY between a head and a tail item is missing, zero, or negative. It is reset to 1.

Examples

This section illustrates how PROC MIRP can be used to calculate inventory replenishment parameters for various network topologies and inputs. Example 3.1 shows how to create the input data sets and invoke the procedure for a four-echelon supply chain. Example 3.2 then demonstrates how to use the procedure to perform scenario analysis. The scenario analysis, also called *what-if analysis*, is used to understand how different strategic decisions can impact the inventory investment across a supply chain. The strategic decisions include the changes in network structures as well as inputs.

Example 3.1. Four-Echelon Supply Chain

Consider a supply chain with a manufacturing process and a distribution network. Two finished goods, F and G, are being produced in a manufacturing plant P. The finished good F is assembled from components C and B, where two units of C and one unit of B are needed for one unit of F. The finished good G is assembled from components B and A, where two units of B and three units of A are needed to make one unit of G. The lead time from external suppliers to components A, B, and C are two weeks, four weeks, and two weeks, respectively. It takes one week to produce F and two weeks to produce G.

After F and G are completed, they are sent to two distribution centers: F goes to $W1$, while G goes to $W2$. The lead times between the plant P and the two warehouses are two weeks.

The distribution center $W1$ receives orders for F from three retail locations, $R1$, $R2$, and $R3$. When an order is placed to $W1$ from the retail locations, it takes three weeks to receive a shipment at $R1$, one week at $R2$, and two weeks at $R3$. The distribution center $W2$ receives orders for G from two retail locations, $R1$ and $R2$. The lead time is two weeks between $W2$ and $R1$, and three weeks between $W2$ and $R2$.

The retail location $R1$ faces demand for the finished good F with a mean of 4 units and a variance of 25 units per week, and demand for G with a mean of 20 units and a variance of 25 units per week. The retail location $R2$ faces demand for the finished good F with a mean of 5 units and a variance of 36 units per week, and demand for G with a mean of 10 units and a variance of 9 units per week. The retail location $R3$ faces demand for the finished good F only, with a mean of 30 units and a variance of 400 units per week.

All retail locations would like to keep their minimum service levels at 95%, while the other locations require minimum service levels of 50%.

In this supply chain network, there are six physical locations: one manufacturing plant, two distribution centers, and three retail locations. There are three components and two finished goods. In order to use PROC MIRP for this network, you first have to map the network to a "virtual" network where nodes are uniquely defined by the pair of SKU and location. Figure 3.8 shows such a network. There is a total of 12 nodes in the network. Each node is labeled by the name of the product and the location that keeps it. For example, node FP stands for the finished good F stored in plant P.

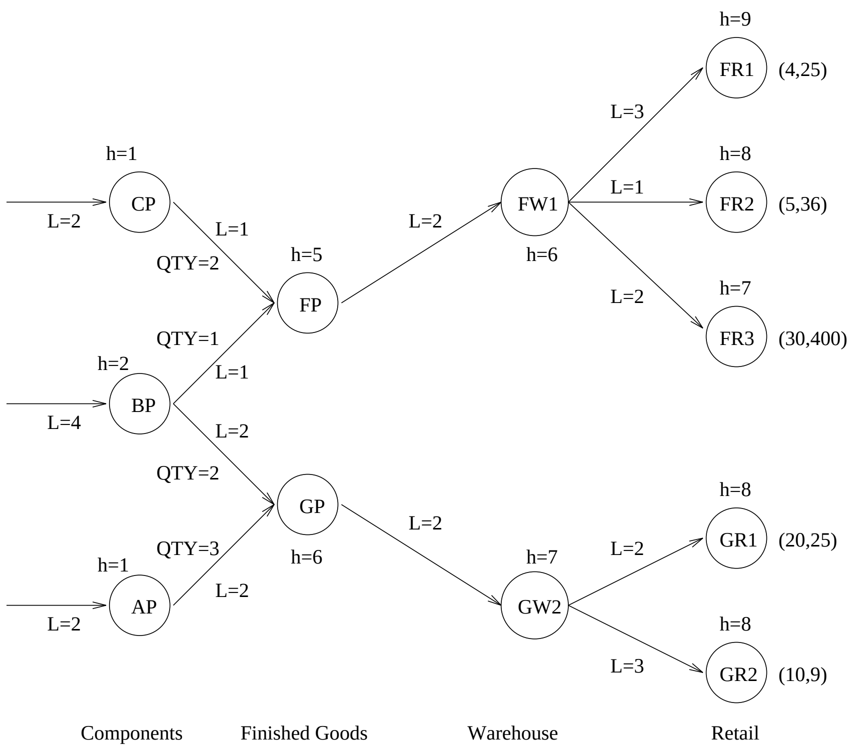


Figure 3.8. Four-Echelon Supply Chain

The following code creates the three input data sets, and then calls PROC MIRP to compute inventory control parameters and other performance measures. Output 3.1.1 shows the resulting output data set.

```
data nodedata;
   input NETWORKID $ ITEMID $ DESCRIPTION $16.
         LEADTIME SERVICELEVEL HOLDINGCOST;
   datalines;
N1 CP component C       2 0.5 1
N1 BP shared component  4 0.5 2
N1 AP component A       2 0.5 1
N1 GP finished good G   2 0.5 6
N1 FP finished good F   1 0.5 5
N1 GW2 warehouse W2     2 0.5 7
N1 FW1 warehouse W1     2 0.5 6
N1 GR2 retailer 2       3 0.95 8
N1 GR1 retailer 1       2 0.95 8
N1 FR3 retailer 3       2 0.95 7
N1 FR2 retailer 2       1 0.95 8
N1 FR1 retailer 1       3 0.95 9
;
```

```
data arcdata;
   input NETWORKID $ HEAD $ TAIL $ QTY;
   datalines;
N1 EXTERNAL CP 1
N1 EXTERNAL BP 1
N1 EXTERNAL AP 1
N1 AP   GP 3
N1 BP   GP 2
N1 BP   FP 1
N1 CP   FP 2
N1 GP   GW2 1
N1 FP   FW1 1
N1 GW2 GR2 1
N1 GW2 GR1 1
N1 FW1 FR3 1
N1 FW1 FR2 1
N1 FW1 FR1 1
;

data demanddata;
   input NETWORKID $ ITEMID $ MEAN VARIANCE;
   datalines;
N1 GR2 10 9
N1 GR1 20 25
N1 FR3 30 400
N1 FR2 5  36
N1 FR1 4  25
;


proc mirp NODEDATA=nodedata ARCDATA=arcdata
          DEMANDDATA=demanddata NETWORKCNT=1 OUT=outds;
run;
```

Output 3.1.1. Output Data Set

Obs	NETWORKID	ITEMID	DESCRIPTION	REORDERPOINT	DEMANDMEAN	DEMANDVAR	ORDERMEAN	ORDERVAR	BKLGUPMEAN	BKLGUPVAR
1	N1	AP	component A	271	90	306	90	306	0.0000	0.00
2	N1	BP	shared component	497	99	597	99	597	0.0000	0.00
3	N1	CP	component C	232	78	1844	78	1844	0.0000	0.00
4	N1	FP	finished good F	98	39	461	39	461	21.7834	624.16
5	N1	FR1	retailer 1	54	4	25	4	25	7.6436	139.30
6	N1	FR2	retailer 2	49	5	36	5	36	7.6436	139.30
7	N1	FR3	retailer 3	166	30	400	30	400	7.6436	139.30
8	N1	FW1	warehouse W1	142	39	461	39	461	25.9470	1119.78
9	N1	GP	finished good G	98	30	34	30	34	7.1635	70.11
10	N1	GR1	retailer 1	80	20	25	20	25	2.8675	18.78
11	N1	GR2	retailer 2	56	10	9	10	9	2.8675	18.78
12	N1	GW2	warehouse W2	97	30	34	30	34	6.1759	78.26

Obs	BKLGDOWNMEAN	BKLGDOWNVAR	ONHANDMEAN	ONHANDVAR	OHCOST	PSCOST	SLREQ	SLSUG
1	12.4414	355.46	12.0874	281.59	12.087	180	0.50	0.54429
2	25.3493	1293.67	21.8034	917.05	43.607	792	0.50	0.54458
3	38.4146	3020.10	28.5566	1403.19	28.557	156	0.50	0.54032
4	25.9470	1119.78	14.5614	348.06	72.807	195	0.50	0.54145
5	1.8944	49.12	31.0353	193.88	279.318	108	0.95	0.95005
6	1.5512	42.67	32.0784	183.89	256.627	40	0.95	0.95143
7	1.5193	67.07	69.5495	1171.01	486.846	420	0.95	0.95097
8	22.9309	1253.71	19.5030	627.37	117.018	468	0.50	0.54407
9	6.1759	78.26	5.6735	55.28	34.041	360	0.50	0.56312
10	0.3803	3.67	17.3897	83.52	139.117	320	0.95	0.95464
11	0.2279	1.71	13.3402	48.44	106.722	240	0.95	0.95483
12	5.7350	75.11	5.8123	58.07	40.686	420	0.50	0.56208

It is important to point out that the internal service levels suggested by PROC MIRP are around 55% for most of the internal (i.e., non-customer-facing) items. This is quite contrary to the common practice of many companies to require high service levels at all locations. This issue is addressed in the next example.

Example 3.2. Scenario Analysis: Service Level Requirement

In Example 3.1, recall that the internal service levels suggested by PROC MIRP are mostly around 55%, which may seem surprisingly low. Companies routinely set their internal service levels as high as their external service levels, possibly because their incentive schemes are set based on the service performance of each individual location. However, strategic decisions of this sort could be very costly to the entire network as a whole, as illustrated in this example.

Consider the network and the input in Example 3.1, and then suppose that the minimum service levels required at all internal items are increased from 50% to 95% with a step size of 5%. At each step, the total on-hand inventory cost of the entire network is calculated. After the calculation is done for all steps, the costs are plotted against the service levels.

```
%macro sens(start,end,inc);

   /*flag the items whose service levels are equal to &start*/
   data nodedata;
      set nodedata;
      if servicelevel=&start then change_ind = 1;
      else change_ind = 0;
   run;

   %let sl = &start;
   %let sl_list = ;
   %let cost_list = ;
   %let numvals = 0;
   %do %while (&sl <= &end);
      data nodedata;
         set nodedata;
         /*The service levels of the flagged items are changed*/
         if change_ind = 1 then servicelevel = &sl;
      run;

      /*run PROC MIRP based on the new service level requirement*/
      proc mirp NODEDATA=nodedata ARCDATA=arcdata
                DEMANDDATA=demanddata NETWORKCNT=1 OUT=outds;
      run;

      /*calculate the total on-hand inventory cost*/
      proc sql noprint;
         select sum(ohCost) into :cost from outds;
         quit;

      /*store the total cost and the service levels into strings*/
      %let sl_list = &sl_list * &sl;
      %let cost_list = &cost_list * &cost;
      %let numvals = %eval(&numvals+1);
      %let sl = %sysevalf(&sl + &inc);
   %end;

   /*store the total cost and the service levels in a data set*/
   data sens;
      %do j = 1 %to &numvals;
         servicelevel = %scan(&sl_list,&j,'*');
```

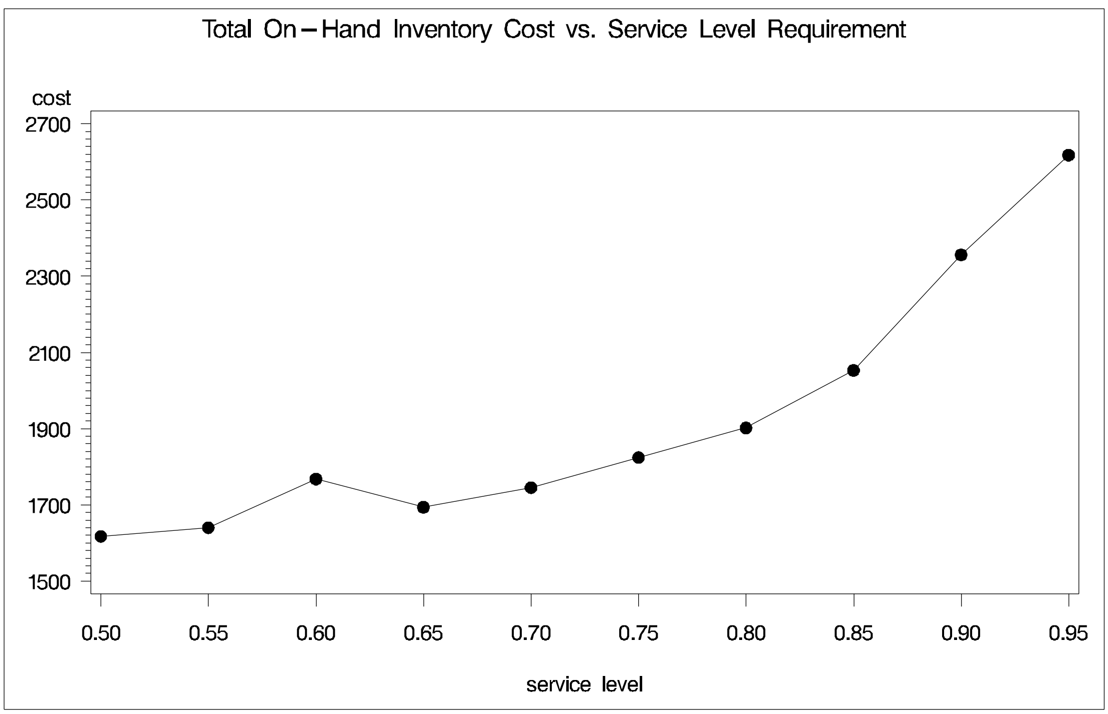
```
            cost = %scan(&cost_list,&j,'*');
            output;
         %end;
      run;

      /*plot the total cost against the service level*/
      symbol interpol=joint;
      axis1 order = (0.5 to 0.95 by 0.05) label = ('service level');
      axis2 order = (1500 to 2700 by 200) label = ('cost');
      proc gplot data=sens;
           symbol color=black;
           plot cost*servicelevel / haxis = axis1 vaxis = axis2;
           title 'Total On-Hand Inventory Cost vs. Service Level Requirement';
           run;

   %mend;

  %sens(0.5,0.95,0.05);
```

Output 3.2.1. Total On-Hand Inventory Cost vs. Service Level Requirement



Output 3.2.1 clearly shows that the total on-hand inventory cost increases as the minimum required service level rises. As the service level approaches 95%, the cost increases very quickly. Comparing the case with 50% service level requirement, the case with 95% incurs at least 60% more inventory cost. This percentage is significant considering the razor-thin profit margins in most manufacturing/distribution/retail businesses. Note that the cost of pipeline stocks doesn't change as the service level changes, because the amount of the pipeline stocks only depends upon the average demand received by the network.

Glossary

A

Average backorder waiting time

The average amount of time that a unit stays in a backordered status. In other words, it is the average time a customer waits for a backordered unit.

Average cost

The average cost (holding and replenishment) incurred per inventory review period. If backorder penalty costs are present, these are included as well.

Average inventory waiting time

The average amount of time that a unit stays in inventory.

B

Backlogging

See *backordering*.

Backorder penalty cost

See *penalty cost*.

Backorder ratio

Average backorders divided by average demand.

Backordering

A standard assumption for many inventory systems such that when there is insufficient on-hand inventory to satisfy a demand, the demand is backordered (backlogged) and satisfied at a later time (rather than lost). See also *lost sales*.

Backordering cost

See *penalty cost*.

Backorders

The total of all outstanding backordered demand. See also *backordering*.

Base lot size

The minimum order quantity. Replenishment orders can only be placed in multiples of the base lot size. See also *(s, nQ) policy*.

Base-stock policy

An inventory policy consisting of a base-stock level, S, and a base lot size equal to 1. The goal of this policy is to maintain the base-stock level. In other words, when the inventory position falls below the base-stock level, S, an order is placed to bring the inventory position to S. This policy is a special case of an (s, S) policy where $s = S - 1$ (or a special case of an (s, nQ) policy where $Q = 1$).

C

Carrying cost

See *holding cost*.

Coefficient of variation

A statistical measure equal to the standard deviation divided by the mean.

D

Decision variables

Quantities that are systematically determined by an algorithm.

Deterministic

A quantity that is known and fixed.

E

Economic lot size

See *economic order quantity*.

Economic order quantity (EOQ)

A type of fixed order quantity, which determines the amount of an item to be purchased or manufactured at one time. The intent is to minimize the combined costs of acquiring and carrying inventory. The basic formula is

$$\text{Quantity} = \sqrt{2dc/h}$$

where d = annual demand, c = average ordering cost, and h = annual inventory holding cost.

F

Fill rate

A service measure that indicates the fraction of demand that is satisfied from on-hand inventory.

Fixed ordering cost

See *ordering cost*.

H

Holding cost

The cost of holding inventory, which may include the opportunity cost of money invested, the expenses incurred in running a warehouse, handling and counting costs, and the costs of special storage requirements, deterioration of stock, damage, theft, obsolescence, insurance, and taxes. Also called "carrying cost."

I

Inventory on order

The total of all outstanding replenishment orders.

Inventory position

A quantity used to measure the current inventory level. It is equal to the sum of on-hand inventory and inventory on order minus backorders.

Inventory ratio

Average inventory divided by average demand.

Inventory ready rate

See *ready rate*.

Inventory turnover

The number of times that inventory "turns over," or cycles during a single review time period. Usually calculated by dividing average demand by the average inventory level.

L

Lead time

The interval of time between when a replenishment order is placed and when it is received.

Lead-time demand

The amount of demand that occurs during lead time.

Long-run average

Given a stochastic process $\{A(t) : t \geq 0\}$, the long-run average $\bar{A}$ is defined by

$$\bar{A} = \lim_{T\to\infty} \frac{1}{T} \int_0^T A(t)dt$$

Lost sales

A standard assumption for some inventory systems such that all unsatisfied demand is lost rather than backordered. See also *backordering*.

Lot size

See *base lot size*.

M

Min-max policy

See (s, nQ, S) *policy*.

O

On-hand inventory

The amount of inventory that is physically in stock.

Optimal policy

The inventory replenishment policy that minimizes the average cost — consisting of ordering, holding, and backorder penalty costs.

Order-up-to level

The target inventory level. Often referred to as the "max" of a min-max policy. See also (s, S) *policy*.

Ordering cost

The fixed cost incurred every time that a replenishment order is placed. This cost includes the expense associated with processing the order and is usually independent of the order quantity.

P

Penalty cost

The cost incurred when a stockout occurs. This cost may include the cost of emergency shipments, cost of substitution of a less profitable item, or the cost of lost goodwill.

Periodic review

Periodic review indicates that the inventory position is known only at discrete points when physical stock counting occurs and replenishment orders can be placed only at these times. Inventory is evaluated periodically (say, monthly) to determine if a replenishment order needs to be placed. The interval of time between two review points is called the review time period.

Poisson process

A stochastic process used to model the arrival pattern of customer transactions. For a precise definition, see P. H. Zipkin's textbook, *Foundations of Inventory Management*, listed in the "References" section in Chapter 2, "The IRP Procedure."

R

Ready rate

The probability that the on-hand inventory at the end of a review time period is positive.

Reorder level

The inventory level at which a replenishment order should be placed. Often referred to as the "min" of a min-max policy. See also (s, nQ, S) *policy* and (s, nQ) *policy*.

Replenishment cost

See *ordering cost*.

Replenishment cycle

The interval of time between two successive replenishment orders.

Review time period

The interval of time between two successive inventory reviews. See also *periodic review*.

Review-time demand

The amount of demand that occurs during a single review time period.

S

(s, nQ) *policy*

An inventory policy consisting of a reorder level, s, and a base lot size, Q. When the inventory position falls to or below the reorder level, s, an order is placed to bring the inventory position just above s. The size of this order is a multiple of the base lot size, Q. In other words, if the inventory position is y, and $y \leq s$, then an order of size nQ is placed, where n is the smallest integer such that $y + nQ > s$.

(s, nQ, S) *policy*

An inventory policy consisting of a reorder level, s, and an order-up-to level, S. When the inventory position falls to or below the reorder level, s, an order (in multiples of Q) is placed so as to bring the inventory position to or above the order-up-to level, S. In other words, if the inventory position is y, and $y \leq s$, then an order of size nQ is placed, where n is the smallest integer such that $y + nQ \geq S$. The base lot size, Q, is specified by the supplier. If $Q = 1$, the (s, nQ, S) policy reduces to the well-known (s, S) policy, which is sometimes referred to as the *min-max* policy. Note that the size of the replenishment order is always greater than or equal to $S - s$.

Service measure

A metric for measuring policy performance. Fill rate, ready rate, and backorder ratio are common service measures.

Shifted Poisson distribution

A random variable Y is said to have a shifted Poisson distribution ("shifted by A") if A is a nonnegative constant, X is a random variable having Poisson distribution, and $Y = X + A$.

Shortage cost

See *penalty cost*.

Stockout cost

See *penalty cost*.

T

Target service level

The value indicating the desired level of a service measure.

Turnover

See *inventory turnover*.

Index

Your Turn

If you have comments or suggestions about *SAS® Inventory Replenishment Planning 9.1.2 User's Guide,* please send them to us on a photocopy of this page or send us electronic mail.

For comments about this book, please return the photocopy to

SAS Publishing
SAS Campus Drive
Cary, NC 27513
E-mail: **yourturn@sas.com**

For suggestions about the software, please return the photocopy to

SAS Institute Inc.
Technical Support Division
SAS Campus Drive
Cary, NC 27513
E-mail: **suggest@sas.com**